TECHNICAL REPORT
NATICK/TR-10/007

AD _____

DEVELOPMENT OF A CHARACTER SIMULATOR FOR BATTLEFIELD VIRTUAL ENVIRONMENTS PHASE II

by
R. D. Eisler*
A. K. Chatterjee*
M. N. West**
R.M. Beecher***
and
G. Tyra****

***Mission Research Corporation**
Laguna Hills, CA 92653-1327

****** West Engineering, Inc.**
Santa Ana, CA

******* Beecher Research Company**
Dayton, OH

******** SA Technologies**
Poway, CA

April 2010

Final Report
March 1998 – September 2001

Approved for public release; distribution is unlimited.

Prepared for
U.S. Army Natick Soldier Research, Development and Engineering Center
Natick, Massachusetts 01760-5020

DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
21-04-2010	Final	March 1998 – September 2001

4. TITLE AND SUBTITLE

DEVELOPMENT OF A CHARACTER SIMULATOR FOR BATTLEFIELD VIRTUAL ENVIRONMENTS – PHASE II

5a. CONTRACT NUMBER
DAAN02-98-C-4012

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

R. D. Eisler*, A. K. Chatterjee*, M. N. West**, R.M. Beecher***, G. Tyra****

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Mission Research Corporation
23052 Alcalde Drive, Suite A
Laguna Hills, CA 92653-1327

8. PERFORMING ORGANIZATION REPORT NUMBER

MRC-COM-R-01-0496

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Natick Soldier Research, Development and Engineering Center
ATTN: RDNS-TSM (T. Gilroy)
Kansas Street, Natick, MA 01760-5020

10. SPONSOR/MONITOR'S ACRONYM(S)
NSRDEC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NATICK/TR-10/007

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

* Mission Research Corporation, Laguna Hills, CA 92653-1327
** West Engineering, Inc., Santa Ana, CA
*** Beecher Research Company, Dayton, OH
**** SA Technologies, Poway, CA

14. ABSTRACT

Report developed under Small Business Innovation Research contract. This Phase II SBIR effort initiated development of a software package that contains a digital anatomy that can be scaled to fit laser-scanned contours of a male soldier. The anatomical model can be articulated and used as stand-alone software or employed as a character simulator that interacts with virtual environments. Protective equipment with different coverage areas and designs can be incorporated onto the soldier. Tissue damage from various forms of battlefield trauma including penetrating wounds from fragmenting munitions, flechettes, and bullets and blunt trauma from non-penetrating projectiles and blast can be assessed. Currently two different anatomical models are included and can be selected by the user – a non-proprietary model based on the dissections of Eyclechymer and Shoemaker and a model based on the National Library of Medicine's Visible Human Project and the proprietary segmentation conducted by Gold Standard Multimedia™. The user can additionally select from two types of projectile-tissue retardation algorithms in the wound ballistic analysis – the retardation algorithms and coefficients used by ARL in its ComputerMan code or those developed by MRC under DARPA sponsorship for battlefield trauma virtual surgery simulators. The character simulator has been nominally designed to interface with an urban warfare virtual environment under development by MRC for STRICOM in another Phase II SBIR.

15. SUBJECT TERMS

MODELS	SIMULATION	HUMAN BODY	WEAPONS EFFECTS	ANATOMICAL MODELS
COMBAT	SMALL ARMS	BATTLEFIELDS	WOUND BALLISTICS	WOUNDS AND INJURIES
TRAUMA	SIMULATORS	SBIR REPORTS	CASUALTY ANALYSIS	COMPUTER PROGRAMS
SCANNING	PROBABILITY		VIRTUAL REALITY	VIRTUAL ENVIRONMENT
COMPUTERIZED SIMULATION				

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Tom Gilroy
U	U	U	SAR	132	19b. TELEPHONE NUMBER *(include area code)* (508) 233-5855

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page intentionally left blank

DEVELOPMENT OF A CHARACTER SIMULATOR FOR BATTLEFIELD VIRTUAL ENVIRONMENTS – PHASE II

INTRODUCTION

This Phase II Small Business Innovation Research (SBIR) effort initiated development of a software package that contains a digital anatomy that can be scaled to fit laser-scanned contours of a male soldier. This US Army Natick Soldier Research, Development and Engineering Center funded effort ran from March 1998 to September 2001. The anatomical model can be articulated and used as stand-alone software or employed as a character simulator that interacts with a virtual environment. Protective equipment with different coverage areas and designs can be incorporated onto the soldier. Tissue damage from various forms of battlefield trauma including penetrating wounds from fragmenting munitions, flechettes, and bullets and blunt trauma from non-penetrating projectiles and blast can be assessed. Currently two different anatomical models are included and can be selected by the user – a non-proprietary model based on the dissections of Eyclechymer and Shoemaker[1] and a model based on the National Library of Medicine's Visible Human Project and proprietary segmentation conducted by Gold Standard Multimedia™. The user can also select from two types of projectile-tissue retardation algorithms in the wound ballistic analysis – the retardation algorithms and coefficients used by Army Research Laboratory (ARL) in its *ComputerMan* code or those developed by MRC under DARPA sponsorship for battlefield trauma virtual surgery simulators. The character simulator has been nominally designed to interface with an urban warfare virtual environment under development by MRC for STRICOM in another Phase II SBIR.

1.1 PROGRAM MOTIVATION

In virtually all work places and military operations, humans are exposed to hazards that can produce injury or in some cases are fatal. Tremendous resources are invested annually to mitigate these hazards and their effects on people. These efforts often require fabricating and testing protective equipment and workplace environments. These procedures are expensive, involve long timelines, and are not easily extrapolated to settings other than those actually tested.

Further, equipment must be designed to act in concert with other components of the soldier system to ensure maximum gains from system synergism and not degrade soldier physical performance. Currently, this is accomplished through a regime of static analysis, testing, and physical mock-ups with the equipment in many cases de-coupled from the soldier and other basic personal equipment up until the time that equipment prototypes are field-tested. The capability developed in this effort will not only enable detailed static analysis but an assessment of how the

[1] Eyclechymer, A.C., and Shoemaker, D. M., A CROSS-SECTION ANATOMY, D. Appleton and Company, New York, 1911.

equipment interacts with the soldier under battlefield conditions, other basic personal equipment, and how the equipment contributes to soldier survival in battlefield environments very early in the design cycle.

This capability brings computer aided design tools typically used to design other military assets to the soldier protective equipment design community as well. In addition to its importance as a design tool, the software is valuable as a training tool for ground combat soldiers and military police. Because of the high-risk nature of the battlefield, the training environment for soldiers is often divorced from the hazards and associated stress that can disrupt decision-making processes.

Decision-making can be thought of as including four conceptual steps: (1) Orienting to the stressful environment, (2) Observing, (3) Deciding, and (4) Acting. Training personnel often refer to this sequence of steps by the acronym OODA (Orienting, Observing, Deciding, and Acting). As a consequence, the training environment is usually designed to hone only specific skills, usually just the "acting" part of the process, as opposed to the whole decision making procedure in a "high risk" stressful environment. For example, firearm proficiency is often attained by training on a static target range. It is well known by the military however that there is little relationship between marksmanship on a target range and firearm accuracy in combat. In surveys conducted by the military, for example, the distribution of striking locations on battlefield casualties from aimed fire is seen to be completely random (however slightly biased toward the head and neck regions) and closely resembles patterns from fragmenting munitions. By creating a virtual environment in which the trainee may be completely immersed, all aspects of tactical decision-making may be exercised.

In a similar vein, capabilities and design features of notional equipment and workspaces can be assessed, interactively changed, and comprehensively studied in a virtual environment as an initial step (as opposed to one of the last steps as is currently the case) prior to fabricating prototypes. This will lead to more robust protective equipment and safer workplace designs that are developed more quickly, and at less cost.

The virtual prototyping capability developed in this effort will enhance the design of protective equipment in several new and innovative ways as well as provide additional capabilities to the Army that can be exploited in training, mission rehearsal, operational scenario analysis, and the design of notional equipment other than protective equipment. The most interesting capability is perhaps provided by the visualization system that will serve to integrate end users at the beginning of the design process. This promotes implementation of concurrent engineering to all interested parties at all levels of the design process.

1.2 PROGRAM OVERVIEW

In addition to providing enhanced models of protective equipment performance the proposed effort would develop a digital human that could interface with the protective system models.

1.2.1 Digital Human Model

A digital soldier model was developed with accurate body contours, anthropometry, and internal anatomy. Body contours are based on highly resolved 3D models of actual soldiers obtained from body scans at Natick using a Cyberware® body scanner. External landmarks were related to internal anatomical features that were used to map anatomical data onto the 3D body model. This anatomical data was obtained from the National Library of Medicine's Visible Human project and MRC's existing MRCMAN model based on a 3D reconstruction of the Eyclechymer and Shoemaker cross sectional data in Reference 1.

Currently, anthropometry data is based on linear measurements of fiducials on the body surface. These dimensions are then allocated to a database describing different percentile classes. In this way, for example, a 10, 50, and 90% percentile body can be obtained. Mannequins with exchangeable parts representing different percentile classes can then be used for "mock-ups" and assessing prototype fit. This approach has various shortcomings however. First, existing anthropometric databases do not adequately account for *somatype*, that is, breadth, depth, and contour data and its correlation with the linear dimensions typically collected. Second, joint locations and centers of joint rotation relative to external landmarks are not known with adequate precision. Finally, there is no high-resolution, integrated capability that explicitly models the relation between internal anatomy and external body surfaces. Currently this is done using intuition and is clearly not optimal.

1.2.2 Fatigue, Trauma, and Injury

The digital human discussed above interfaces with models of movement rate, cardiovascular fatigue, and heat stress available in the Integrated Unit Simulation System (IUSS)[2]. Additionally, the digital human incorporates models of tissue damage from penetrating wounds and blunt trauma. Animated movement, collision detection, and spatial reasoning are provided by scripted software developed using the Motivate™ package from the Motion Factory.

Models of penetrating wounds incorporated into the current simulator were developed by MRC for virtual surgery simulators developed by the Advanced Biomedical Technology program at DARPA.[3,4,5] This effort was a Phase III follow-on to MRC's Phase II development for Natick of

[2] IUSS is being developed by Simulation Technologies, Inc. (STI) in Dayton, Ohio under contract to Natick. MRC is a subcontractor to STI to assist in development of ballistic casualty modules for IUSS.

[3] R.D. Eisler, *Analytical Simulation of Wound Tracts from Missile Penetration*, PROCEEDINGS OF SEVENTH INTERNATIONAL SYMPOSIUM OF WEAPON TRAUMATOLOGY AND WOUND BALLISTICS, St. Petersburg, Russia, 20-23 September 1994 (Unclassified).

[4] R.D. Eisler, A.K. Chatterjee, and G.H. Burghart, *Simulation and Modeling of Penetrating Injuries from Small Arms*, published in: HEALTH CARE IN THE INFORMATION AGE: FUTURE TOOLS FOR TRANSFORMING MEDICINE, presented at the Medicine Meets Virtual Reality 4 International Symposium sponsored by The University of California School of Medicine, the Advanced Research Projects Agency, American Psychiatric Association, Institute for Telemedicine, Society for Minimally Invasive Surgery, Society of Gastrointestinal Endoscopic Surgeons, and Society of Cardiovascular and Interventional Radiology, San Diego, California, 17 – 20 January 1996 (Unclassified).

[5] A.K. Chatterjee, R.D. Eisler, and G.H. Burghart, *Ballistic Penetration into Gelatin*, IMPACT, WAVES, AND FRACTURE, Proceedings Of The Werner Goldsmith Symposium, sponsored by the Applied Mechanics Division of

the *Soldier Protective Ensemble Computer Aided Design* (SPE/CAD) system which includes the MRCMAN model.[6] Blunt trauma in the thorax and abdomen is assessed in the current character simulator using two approaches. First, a viscous dashpot model of the lungs[7] developed by the Lovelace Institute and resurrected by Larry Josephson at the Naval Weapons Center at China Lake has been incorporated. This model describes intrathorasic pressure, chest wall displacement, velocity, and acceleration given a prescribed pressure time or displacement time-history on the surface of the chest. For non-penetrating blunt impact on body armor, validated models of rear surface PASGT armor displacement time-histories have been developed by MRC and can be used as initial conditions for the viscous dashpot model.[8] Additionally for both abdominal and thoracic blunt trauma, the armor displacement-time histories can be used directly with the Viano viscous criterion[9] to predict blunt trauma injury to these body regions. This criteria however has only been validated for blunt trauma associated with pressure time-histories lasting tens of milliseconds and encompassing large areas of the body (i.e., car crash scenarios). Whereas this approach is probably valid for kinematic trauma (i.e., where something falls on the soldier or the soldier is knocked against something) its application to ballistic impact remains uncertain. Part of the difficulty is the lack of medical case histories or anecdotal information that indicates the type of tissue damage associated with injury in these scenarios.

1.2.3 Motion and Articulation of the Digital Human Model

The character simulator incorporates segment mass properties and joint equations of motion obtained from the Articulated Total Body Model (ATB)[10]. This allows external forces from weapon effects to evoke motion and displacements in the body for trauma assessment. The digital human however functions in this environment in a relatively passive manner unlike JACK that has the ability to do work in the environment for ergonomic evaluations. While incorporating the necessary biomechanical models into the proposed digital human model might be desirable for a virtual prototyping effort in order to assess task performance, two trade-offs mitigate against this. First, the requisite biomechanical models and input data are very immature and to some extent not validated. Second, the incorporation of these models imposes a massive computational burden making these models unsuitable for real-time simulation in a virtual environment.[11]

---

the American Society of Mechanical Engineers and the University of California at Los Angeles, Los Angeles, 28 - 30 June 1995, ASME Applied Mechanics Division, Volume 205, pp. 9-20, 1995 (Unclassified).

[6] R.D. Eisler, *Soldier Protective Ensemble Computer Aided Design (SPE/CAD) System: An Evolving Tool for the Evaluation of Battlefield Protective Equipment*, PERSONAL ARMOUR SYSTEMS SYMPOSIUM, Colchester, England, 21-25 June 1994 (Unclassified).

[7] L.H. Josephson and P. Tomlinson, P., "Predicted Thoracic-abdominal Response to Complex Blast Waves," THE JOURNAL OF TRAUMA, Vol 28, No. 1 Suppl, pp. S116-S124, January 1988.

[8] R.D. Eisler, et al, *Improved Fibers and Material Systems for Personnel Ballistic Armor*, Final Report for US Army Natick contract DAAK60-93-C-0024, Mission Research Corporation Report MRC-COM-94-R-0454, Costa Mesa, California, October 1994 (Unclassified).

[9] D.C. Viano, and I.V. Lau, "Biomechanics of Impact Injury," General Motors Publication GMR-6894, Warren, MI, 13 December 1989.

[10] L.A. Oberfell, et al, *Articulated Total Body Model Enhancements*, Volume 2: User's Guide, Armstrong Aerospace Medical Research Laboratory report AAMRL-TR-88-043, January 1988.

[11] Undocumented conversation on 17 December 1996 with Callis Goodrich at NRaD in San Diego and Chris Field a contract employee at NRaD supporting MODSAF development, the NRaD MODSAF simulation facility had to replace JACK with the Boston Dynamics *Dismounted Infantryman* model to address this problem.

Problems with the biomechanical approach for animating motion are at least threefold. First, there are no models that accurately characterize accommodation to a design if two or more body dimensions are critical. In fact, data relative to range of motion in other than principal body planes, particularly involving combinations of joint motions and independent axial rotations are almost non-existent. Second, human muscle strength is highly idealized. It is usually measured under static ("isometric") conditions (i.e., body does not move and muscles remain at same length) that is only weakly related to dynamic exertion of strength. Alternatively, human muscle strength may be idealized as isotonic (i.e., muscle strength remains constant) which is equally unrealistic. Finally, the interaction of the protective equipment with the body surface under dynamic conditions (that is while the body moves) is currently analytically intractable. Given the tremendous variety of human sizes, shapes, performance capabilities and limitations, analytical models entailing the capabilities above are of limited utility at the present time and do not argue for incorporation and the necessary investment of computational resources which would compromise performance of other capabilities.

Animation in the visualization system was achieved by using the Motivate™ software package. *Motivate* is a software package used by movie animators and provides spatial reasoning and allows "training" of characters to perform tasks involving articulation of joints. Motivate generates script files for these various tasks.

1.2.4 Visualization System and Software Architecture

The design of protective equipment as conceived of in this effort has three basic cycles: static design, dynamic design, and *man-in-the-loop* testing. In each of these phases, the application of 3D visualization is used to improve the design process.

During static analysis, various factors are evaluated and engineering trade-offs are made. The design of personal protective equipment must balance factors of wearability, durability, and overall effectiveness of protection. Successful completion of the proposed Phase II effort will provide a significant increase in the amount and quality of the data available to the designer. To make effective use of this influx of data, the designer needs improved tools to evaluate the results of trade-offs made during each design iteration. Using 3D modeling and rendering as initiated in this effort provides a foundation for such a tool set.

Wearability is a function of several factors such as mass distribution, abrasion, heat transfer, and flexibility. The 3D image provides the designer with color-coded representations of how the proposed equipment interacts with a representative selection of wearers. Given the current and short term analysis capability, the main factors to be evaluated at this stage are reduction in joint moment generating capability, increased work load during movement, stability on different terrain, mass distribution, total weight, thermal load, and casualty reduction. As analytical models of human mobility and range of motion improve, these factors can be incorporated into this analysis module.

Combining analytical models of mobility and range of motion with material property data will

allow evaluation of protective equipment for durability. Improved durability can be obtained by identification, as early in the design cycle as possible, of points of high stress or wear. This information permits the designer to select materials and construction techniques with greater value and performance at a lower cost.

The overall effectiveness of personal protective equipment is a balance of the greater encumbrance imposed by the equipment versus the higher probability of mission completion through reduced casualties. Here again, 3D imagery can be used to assist in the design process. This is accomplished by identifying the range of threat environments in which the equipment is to be used, and the distribution of injury severity and type which can be expected in those environments. This is then translated into a visual model indicating areas of the body at greatest risk to different types of injury (i.e., penetrating wounds and blunt trauma). The designer can then interactively apply protective material, in a selective manner, until an acceptable risk level is reached. The designer can thus use a minimum of mass, thereby reducing encumbrance, while still providing the maximum level of protection at a given cost point.

Once a cycle of static design is competed, a dynamic analysis is needed to validate the conclusions of the static analysis. The dynamic analysis looks for such features as significant changes in bulk which would affect the selection of cover or access to confined spaces. Significant changes to endurance, due to better heat transfer or reduced mass, could effect the selection of tactics. The dynamic analysis would take synthetic actors/combatants through a series of scenarios. Each run would be scored against a baseline simulation. The simulation would be controlled by a combination of self-directed characters and a human director who could make use of new tactical opportunities.

The static analysis provides the designer with statistical data on the type and distribution of impacts from which the equipment is suppose to protect the wearer. The dynamic analysis will demonstrate that the original threat model is consistent with the manner in which the equipment is used. The results of these tests would be passed back to the static analysis to provide a more complete analysis of the protective equipment.

When the design is approaching the point where prototype equipment is to be built, a virtual environment can be used as a final human factors check of the protective equipment. This serves as an intermediate step between theoretical analysis and actual field testing. At issue is the response of teams using the new equipment.

Given a threat environment, a team using baseline equipment would normally select one of several tactical solutions to resolve the situation. Since no solution to a combat situation is risk free, every solution must be evaluated on a statistical basis, weighing team losses against mission success. By substituting the proposed protective equipment, new tactical options can be evaluated as well as changes to loss rates using conventional tactics.

By running a number of simulations in a virtual environment, the evaluators can quickly develop an appreciation of the new equipment's strength and limitations, even before the prototypes have been fabricated. By the time actual field testing begins, the group doing the evaluation has had an

opportunity to develop a more comprehensive set of tactics to apply. This should result in shorter, more effective field testing. A unique advantage of this visualization system is that it provides a common basis in which to integrate Army field officers early into the design process taking concurrent engineering to its logical conclusion.

1.2.5 Anthropometry and Anatomical Modeling

A key element of the Virtual Prototyping tool is biofidelity. Biofidelity refers to the biological realism of the mathematical, or computer, model of the human body. In the past, available technologies were at best limited to developing prototype human-use equipment or workspaces in a two-step process. First, information on human measurements was analyzed to create ranges representative of the variety of sizes and shapes thought necessary for accommodation. Second, physical prototypes were created for those size/shape ranges. For instance, in creating headform sizes, linear measurements of servicemen's heads, such as length, breadth, and circumference, were analyzed to find the minimum and maximum ranges that would accommodate 95% of all servicemen. Then, the ranges were divided into a series of steps that defined the sizes. Finally, physical headforms were created that had the defined head measurements. The biofidelity of the headforms was limited by the relatively few linear measurements used to specify their dimensions. Aspects of head shape not covered by the measurements were not part of the specification and the artist creating the headforms was free to interpret.

A method using more shape-related information, and thus more biofidelic, was used to create headforms for helmet sizing.[12] Here, movable probes in a shell over the subject's head were adjusted to record clearance at regular intervals. This information was also used to create physical headforms.

Depending upon the application – and availability of modern computers – full-body modeling has followed different paths in the biofidelity of prototype development. In clothing and body protective equipment design, where sizing tends to follow traditional apparel industry models, data documenting the size and shape of military personnel was practically ignored until relatively recently. Civilian sizing still incorporates little or no body measurement data in design. Several early reports from Army anthropologists discuss the availability of detailed anthropometric surveys useful for design applications[13,14,15] and then despair that the information is not used.[16] Designers were using traditional clothing industry methods and sizes to produce prototype articles which were then tariffed with little regard for the size/shape distribution of the customers. This practice changed in the last decade when available anthropometric data was used

[12] W. Claus, L. McManus, and P. Durand, "Development of Headforms for Sizing Infantry Helmets", TR 75-23-CEMEL, US Army Natick Laboratories, 1974.
[13] Hooton, E. A., "Album Illustrating Body Build in Relation to Military Function in a Sample of the United States Army", Final Report Contract W44-109-QM-1078, 1948.
[14] Randall, F.E., "Applications of Anthropometry to the Determination of Size and Clothing", Environmental Protection Series Report No, 133, U.S Army Quatermaster Climatic Research Laboratory, 1948.
[15] White, R. M., "A Summary of Present Research in Army Anthropometry", Am. J. Phys. Anthrop. n.s. 8:272, 1950.
[16] White, R. M., "Some Applications of Physical Anthropology", J. Wash. Acad. Sci. 42(3):65-71, 1952.

to produce sizing schemes and prototypes were systematically fit-tested on sample populations representative of the forces to be outfitted.[17]

Physical human models – anthropomorphic mannequins or crash test dummies – have been used to test prototype equipment, especially where there is some physical danger as in high-acceleration tests of aircraft escape systems. The biofidelity of these models has varied but the aim of the designers has been to duplicate overall size and weight, not the details of joints, body segments, and internal structures. Detailed evaluation of state-of-the-art automobile mannequins showed that there was little similarity to any statistically representative human data model.[18] From these evaluations and taking advantage of all available 3D human body geometric, static, and inertial data, a series of mannequins was designed with as much biofidelity as possible, the *Advanced Dynamic Anthropomorphic Mannequins* (ADAM). Testing environmental conditions has also led to the construction of mannequins instrumented to measure conditions of environmental stress such as heat/cold and humidity.

Prototyping workspaces in military vehicles, such as aircraft cockpits, traditionally involved using a small sample range of a very few anthropometric dimensions. On some aircraft, such as the T-38, two subjects, one large, one small, were used to evaluate accommodation in the cockpit.[19] While the biofidelity of the subjects would be unquestioned, the representation of flying personnel by two people (white males) left out a large amount of the variability in size and shape. More recently, physical prototypes have been subject to evaluation using more sophisticated statistical models of human size and shape distribution.

Workspaces are also being prototyped using computer models of both the space and personnel. Earlier models such as the *COMBIMAN* [20] had very simple representations of the workspace. The human model was represented by geometric shapes (cylinders, ellipsoids) for body segments, but it was based on available anthropometry (a 1968 survey of Air Force flying personnel) and so had reasonable biofidelity. The model was intended to test cockpit reach and vision, but performance was not realistic or strongly correlated with real human performance tests. In a technical sense, *COMBIMAN* was in part succeeded by *Crew Chief.*[21]

Crew Chief was developed as part of the Air Force Computer-Aided Acquisition and Logistical Support (CALS) program called DEPTH which sought to have computer prototypes of acquisitions which could model human-machine interactions in such areas as maintenance and performance. *Crew Chief* embodied a new approach to man modeling by incorporating performance data into the model. Extensive tests of reach, vision, and strength on a sample of

---

[17] Gordon, C. C., "Anthropometric Sizing and Fit Testing of a Single Battledress Uniform for U.S. Army Men and Women", in "Performance and Protective Clothing", ASTM STP 900, R. Barker & G. Coletta, eds., pp.581-592, 1986.

[18] I. Kaleps, R. White, R. Beecher, J. Whitestone, and L. Obergefell, "Measurement of Hybrid III Dummy Properties and Analytical Simulation Data Base Development", AAMRL-TR-88-005, Armstrong Lab., WPAFB, 1988.

[19] G. Zehner, personal communication.

[20] F. Bates, S. Evans, H. Krause, and H. Luming, "Three Dimensional Display of the COMBIMAN Man-Model and Workspace", AMRL-TR-74-15, WPAFB, 1974.

[21] M. Korna, et al., "User's Guide for Crew Chief: A Computer Graphics Simulation of an Aircraft Maintenance Technician", AAMRL-TR-88-034, 1988.

subjects representing anthropometric ranges in the Air Force were performed, and the results incorporated into the performance aspects of *Crew Chief*. This greatly improved the biofidelity of the model. Rather than depending on *creating* performance in a crude 3D world, the model used a real-world performance database to test the 3D world.

Best known among the human models now used for prototyping and testing is *Jack*, developed and maintained at the University of Pennsylvania. Three dimensional objects in *Jack*, including humans, can be modeled in some detail, including body segments, surfaces, joints, surface characteristics and joint constraints, so that a high degree of biofidelity can potentially be achieved. *Jack* can be both a static and dynamic model. Given the flexibility of the model, internal body structures can also be modeled. The problem is that the representation of surfaces and objects requires much handwork in constructing – one cannot automate a transformation of human 3D data from, say, CAT scans or laser digitizer, to the *Jack* data format. This has inhibited the flexibility of *Jack* in representing real body size/shape distributions. A front-end program called *SAS* is supposed to input anthropometry and output re-scaled *Jack* bodies, but limited experience using SAS tends to indicate some reliability problems.[22]

Another direction in biofidelic human modeling was taken by the Articulated Total Body Model (ATBM), originally developed by the Department of Transportation[23] and now maintained by the USAF Armstrong Laboratory.[24] The ATBM was first and foremost a dynamic model developed to predict human dynamics in automobile crashes. Later applications have included aircraft cockpit and escape dynamics. In order to incorporate sophisticated and computationally intense equations of motion, the objects, including humans, are represented by simple geometric shapes – ellipsoids, cylinders, rectangular solids. This compromises the biofidelity of the human models in terms of surface representation, but permits modeling the dynamics of individual body segments more realistically than in other computer programs. The relatively simple geometric data structures permit easy construction of environments for prototype testing. Enhancing the biofidelity of the human models in the ATBM is GEBODIII, a program which computes body data sets based on actual 3D human body data, including joint locations and performance.[25]

A recent tool that could combine anthropometry, internal geometries, and dynamic properties into a human model useful for battlefield simulations is Mission Research Corporation's *MRCMAN* module of the Soldier Protective Ensemble Computer Aided Design System – (SPE/CAD).[26] Development of this model was sponsored by Natick (DAAK60-92-C-0008) in a Phase II SBIR.

[22] Personal communication from Dr. R.M. Beecher documenting his experience.
[23] J. Fleck, F. Butler, and N. DeLeys, "Validation of the Crash Victim Simulator", DOT-HS-806-280, 1982.
[24] L. Obergefell and T. Gardner, "Articulated Total Body Model Enhancements", AAMRL-TR-88-043, Armstrong Lab., WPAFB, 1988.
[25] M. Gross, "The GEBODIII Program User's Guide and Description", AL-TR-1991-0102, Armstrong Lab., WPAFB, 1991.
[26] R. D. Eisler, "Soldier Protective Ensemble Computer Aided Design (SPE/CAD) System: An Evolving Tool for Evaluation of Battlefield Protective Equipment", MRC-COM-94-R-0477, presented at PASS 94 Symposium, Colchester, UK, 21-25 June, 1994.

The human anatomy in *MRCMAN* consists of 80,000 voxels representing more than 250 tissue types. Nineteen body segments have external geometries composed of ellipsoidal slices. The jointed figure can be interactively placed in any position. Overall anthropometric scaling, which could be used to model representative soldiers, is currently limited. MRCMAN's tremendous advantage for this application is its straightforward data structure for representing body geometries. This data structure could better accommodate increasingly comprehensive human 3D data, such as surface laser and internal MRI scans. Linked to an anthropometry generator, as proposed below, MRCMAN is a good candidate for a flexible human model for virtual prototyping.

Numerous other computer programs are capable of modeling humans in 3D for various limited applications, such as ADAMS/Android, Dench/ERGO, LifeForms, McDonnell Douglas Human Modeling System, ShapeAnalysis, and Musculographics, Inc., Software for Interactive Musculoskeletal Modeling (SIMM). General purpose CAD packages, such as Alias Designer or ProENGINEER might also be adaptable to human modeling.

Previous efforts at prototyping – virtual or physical – have lacked the combination of data, software, 3D recording technology, and high-powered computer resources that are available today. The resources now available and those that are being developed in companion programs can be combined to produce human modeling capabilities with unprecedented biofidelity.

1.2.6 Program Scope

The Phase II effort consisted of the nine tasks listed below.

Task 1 – Development of integrated anatomical and anthropometrically accurate 3D human body model using full body scanner and visible human data

Task 2 – Incorporate mass properties of human body segments and joint equations of motions for kinematic trauma and human body dynamics calculations

Task 3 – Development of enhanced penetrating wound and blunt trauma models

Task 4 – Character simulator animation and collision detection

Task 5 – Virtual Equipment Models

Task 6 – Interface development

Task 7 – HLA Compliance for Soldier Character Simulator

Task 8 – Design and analysis module development

Task 9 – Preparation of Deliverables

Task 1 – Development of integrated anatomical and anthropometrically accurate 3D human body model using full body scanner and visible human data.

For this Virtual Prototyping effort, the human modeling aspect must have the following capabilities: (1) Background anthropometric database from which statistically generated data sets representing intended size and shape ranges of military personnel can be easily and automatically generated; (2) Geometries of internal body structures which can be scaled and organized to fit the anthropometric external geometry; (3) Database of physical properties of body tissues, organs, and segments including force-deflection and inertial properties; (4) Software to input the above resources and output a body model appropriate for the Virtual Prototype modeling software; (5) Software module as part of the entire Virtual Prototyping system which can manipulate the body model, react to external conditions in the virtual environment, including trauma, and interface with the larger Virtual Prototyping software to communicate results of simulations. Figure 1-1 shows a *roadmap* linking together databases and data analyses. Data and software in **bold** are resources available now which can be adopted or modified for the Virtual Prototyping effort.

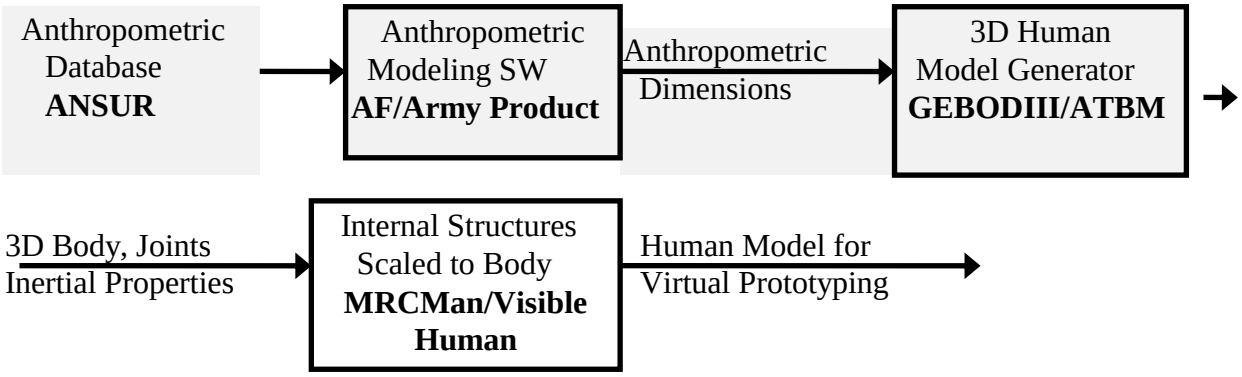

Figure 1-1. Roadmap for human body data processing to produce an input for virtual prototyping

Generating the Human Body Model for Virtual Prototyping

The pathway for generating the human body model is linear with data being added, processed, then the results added to the next step. The process begins with a military anthropometric database from which a data set is generated with dimensions equal to those of the desired human model. Those dimensions, together with information on 3D body surfaces, joints, and body segment inertial properties are then used to produce a 3D body model that is a segmented, jointed shell. The internal geometries data, organs, tissues, fluids, is added and the shell segments size/shape are used to re-scale the internal structures database to fit appropriately inside the body model. The final data set is then formatted for the human model software module in the virtual prototyping software package.

Anthropometric Database

The size and shape distribution of personnel in the armed forces are represented by measurement surveys. Until recently, the only technologies for obtaining measurements were calipers and tapes. Thus, the data is composed almost exclusively of linear measurements – such as lengths, widths, and circumferences. The principal large surveys are of Air Force flying personnel (1968) and Army personnel (1988). The Army Anthropometric SURvey (ANSUR) is the largest and most comprehensive anthropometric survey ever carried out, and was designed to oversample categories of personnel so that future changes in gender, age, and ethnic makeup of the Army could be factored in to update a statistical profile of Army size and shape distribution.[27] Since its publication, other services have carried out so-called mini-surveys which were then matched to ANSUR in order to re-sample ANSUR for statistical analyses. The Natick Anthropology Team maintains a Working Database of ANSUR which can be accessed by others for statistical analysis.

[27] C. Gordon, et al., "1988 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics", Natick/TR-89/044, 1988.

Although the only current large anthropometric databases are composed of linear measurements, the Natick Anthropology Team and the Air Force Armstrong Laboratory have recently acquired whole body 3D laser scanners manufactured by Cyberware Laboratory (Figure 1-2). The Army scanner is being used both for special applications projects at Natick and for a larger scale project sponsored by the Defense Logistics Agency's Apparel Research Network. This latter project will result in the scanning of a large number Army personnel over the next several years. The resulting database will be available for new analyses of true 3D anthropometry. For virtual prototyping, the database will be invaluable in generating state-of-the-art biofidelic human models.

Figure 1-2. Subject being recorded on the Natick whole body laser scanner (courtesy of Brian Corner of Natick)

Figure 1-3 shows an image of a 3D model generated using the Cyberware system using the subject and equipment shown in Figure 1-2. The scanner records surface color as well as surface points at 3 mm resolution. The yellow lines are measurements taken off of the 3D data using software also purchased by the Natick Anthropology Team.[28]

Anthropometric Modeling Software

In developing human models for virtual prototyping, two factors are important. First, each model was to be realistic, or biofidelic. That is, it must have the size/shape of what could be a real person. Second, the set of human models used in any prototyping effort must encompass some predetermined ranges of size and shape within the population. This range is necessary so that the product being prototyped can be said to accommodate some certain percent of the population – usually 90-95%. For many years, anthropometric models were statistically generated to represent a percentile of the distribution of size and shape in the population. Thus, there were often anthropometric data sets said to represent 5th, 50th, and 95th percentile individuals. These data sets were generated using regression equations predicting all anthropometry from a few dimensions at those percentiles - typically stature, weight, and sitting height. The assumption was that this then would result in the accommodation of all personnel between the percentile extremes (here 90%). The problem is that this assumes that there is a linear distribution of size and shape – something which is not true.

This problem began to be addressed about ten years ago by Gregor Zehner of the USAF Armstrong Laboratory in an effort to develop accommodation models for AF cockpit crew station design. Working with Richard Meindl of Kent State University, they developed adapted

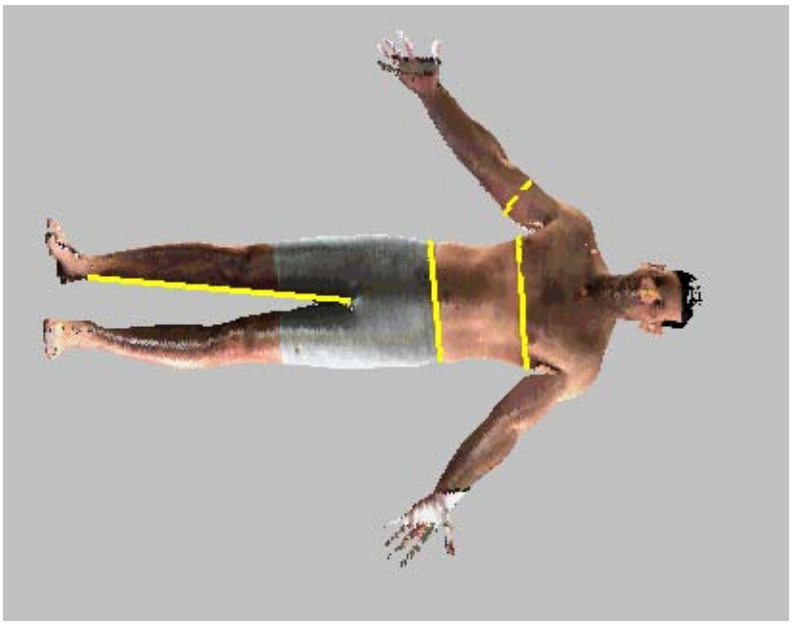

Figure 1-3. 3D Model from Laser Scan of Subject in Previous Figure

[28] Information and Figure 1-2 and Figure 1-3 courtesy Dr. Brian Corner from Natick.

multivariate statistical techniques based on principal components analysis to generate representative anthropometric data sets which accommodated the non-linear size and shape distribution of flying personnel.[29]

Using traditional regression analyses, the user determines which variable, such as stature or sitting height, is most important, and uses that variable to predict the remaining anthropometric dimensions. In principal components, the first step analysis determines which variables, or dimensions in this case, account for most of the variability in the population. Those variables are then used to predict the remaining dimensions in a range of anthropometric data sets which can truly accommodate a specified percentage of the population.

This modeling technique is being generalized in a software package called the Multivariate Accommodation Module (MAM), which is also being evaluated by Brian Corner at Natick. When completed, the MAM will be useful in generating anthropometry from the large surveys. At Natick, this techniques are being applied by Dr. Brian Corner and Dr. Claire Gordon of Natick for *Land Warrior* projects involving helmets, load bearing systems, and modular body armor.

For applications of virtual prototyping, the MAM software or some variant will input anthropometry from the ANSUR Working Database, representing the current population of US Army soldiers, and output anthropometric data sets with the dimensions needed by the next step in the virtual prototyping process – the 3D Human Body Model Generator. The size and shape range of the output data sets will be determined by the user. Thus, in order to accommodate, say, 95% of the population, rather than having three data sets – small, medium, and large – there might be ten sets representing extremes and middles along several principal components.

3D Human Body Model Generator

The final 3D human body model for virtual prototyping will run under a current or successor version of *JOHN O*. (MRCMan). The internal body geometry for *JOHN O*. (MRCMan) is currently described by about 80,000 voxels with tissue properties, while the surface is a series of slices configured from ellipsoids. The body is segmented with joints and can be positioned interactively for use in trauma simulations. The data structures for *JOHN O*. (MRCMan) are not identical to the Articulated Total Body Model (ATBM), but they are similar enough that the software used to generate the body geometry for the ATB can be adapted for the *JOHN O*. (MRCMan) virtual prototyping tool. This data generating software is GEBODIII (Generator of BODy data).

GEBODIII was written by Mary Gross of Beecher Research under contract with the USAF Armstrong Laboratory. It is the only program of its kind to use true 3D human body data – surface geometry, segment inertial properties, and joint locations and constraints – to produce anthropometrically accurate human body models for simulations. The writing of GEBODIII followed several years of research and applications into the uses and modeling of 3D data. The data which was incorporated into the program was previously used for set-piece applications

[29] R. Meindl, J. Hudson, and G. Zehner, “A Multivariate Anthropometric Method for Crew Station Design”, AL-TR-1993-0054, Armstrong Lab., WPAFB, 1993.

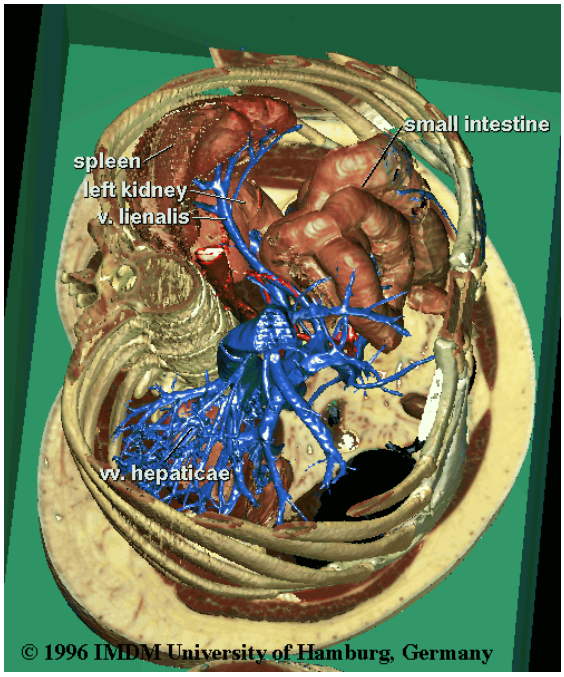

Figure 1-4. Visible Human data of the abdomen modeled using the VOXEL MAN software.

such as the design of new USAF crash test mannequins (ADAM). The output was tested by not only measuring the anthropometry, but putting the body model into action to see if the joints and body proportions and shapes were realistic.

The 3D geometry, joint locations, and inertial properties for GEBODIII were largely based on the program FRNKNSTN, also written by Beecher, which was a static 3D human model. FRNKNSTN inputs full-body surface stereophotogrammetric data sets of men and women, computes joint locations based on surface landmarks, and has the capability to move the subjects either interactively, or through a batch movement file. The data sets also contain inertial properties for the body segments so that whole-body properties, such as center of gravity, can be calculated in any position.

Linking together MAM as anthropometric modeling software and GEBODIII (or a successor) as part of a 3D body model generator is an effort now being undertaken at Armstrong. In addition to the input used by the ATBM, whole body laser scans will also be incorporated and high-resolution body surfaces will be produced. These products will be well-positioned to be used or adapted by the virtual prototyping effort as part of the program to generate a 3D human model for *JOHN O*. (MRCMan).

Anatomical Modeling

The internal anatomy for *JOHN O.* (MRCMan) is based on cross-sectional drawings published early in this century[30] from fifty different supine cadavers and digitized. In stacking the various cross-sections in the digital model, various approximations were necessarily made to achieve alignment and correct posture in a standing position. The level of detail is less than can be obtained from current sources however. The National Library of Medicine's Visible Human Project (VH) has published one full-body male data set and is about to publish a female set. The detail of the data is unprecedented. After being fresh-frozen, the male cadaver was CT scanned and sectioned horizontally in 1 mm thick slices, with each slice photographed at 0.33 mm resolution. The result is a digital data set with 24 bit RGB color, and 0.53 mm CT resolution. The data set is also unprecedentedly large – 15 gigabytes.

While the body is recorded in great detail, the individual tissues and organs are not separated, or segmented, from each other in the raw data. Segmentation is a major hurdle to the efficient use of medical imaging data because it cannot be performed automatically. That is, there is no software

[30] Eyclechymer, A.C., and Shoemaker, D. M., A CROSS-SECTION ANATOMY, D. Appleton and Company, New York, 1911.

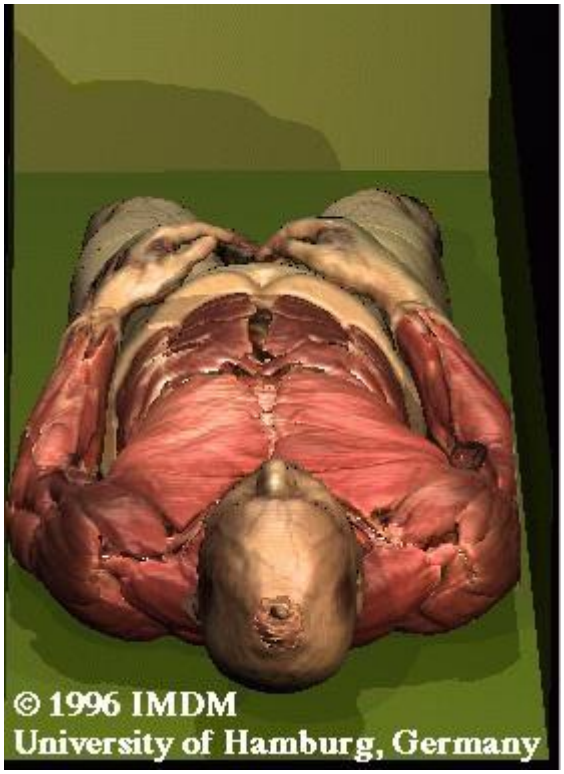

Figure 1-5. Visible Human data on the male torso modeled in the VOXEL MAN software.

which can input a CT or VH data set of, say, the abdomen, and extract, automatically, only those voxels which represent the pancreas. Software which performs segmentation requires significant user interaction.

Fortunately, several organizations are working to produce segmented versions of the VH data. The University of Hamburg (Germany) has a long record of converting medical images to graphics and 3D visualizing. Their current project, VOXEL-MAN is a human model initially based on CT imaging data, but which is now incorporating the VH data.[31] This model is a high-resolution segmented data set with browsing software allowing the user to "slice-and-dice" the data in order to view it from any perspective and at any depth. The segmentation is detailed and includes labels for major structures (see Figures 1-4, 1-5, and 1-6). For the virtual prototyping effort, the importance of this work is that it is available, uses the most advanced data, and it can be adapted to output data for *JOHN O*. (MRCMan).

Currently, VOXEL-MAN is available commercially (Springer-Verlag) as an electronic atlas for the head and neck regions. The remainder of the body using Visible Human data is in preparation. A search on-line has not discovered any other sources for segmented, VH data sets. The virtual prototyping work could incorporate this data as it became available in useful formats.

To incorporate this information, the internal geometry data must first be re-scaled and reshaped to fit inside the body surface and aligned with the segments and joints specified by the Human Body Model Generator described above. This is not just a matter of pushing, pulling, and distorting the data so that it fits inside. While tissues such as connective tissue, fat, and parts of the intestines are malleable, many organs do not vary much in size regardless of the shape of the body segment in which they are contained.

Two kinds of information are useful here. First are the clinical anatomical descriptions in most anatomy texts which localize many major organs with respect to surface landmarks or palpable bony features.

The second is data on the variability of organ size and weight. Clinical descriptions instruct physicians where to locate various organs under the body surface. These descriptions relate

[31] U. Tiede, T. Schiemann, and K. Hoehne, "Visualizing the Visible Human", IEEE Comp. Graphics & App. 16(1):7-9, 1996.

Figure 1-6. Visible Human male head data viewed in the Voxel MAN software

organs to bony features, such as rib numbers, which locate the upper and lower range for where the heart "projects" onto the front of the chest.

The surface features and landmarks necessary to register the internal anatomy can best come from the high-resolution laser digitized subjects which will be recorded and analyzed on Cyberware whole body scanners, such as the one located with the Anthropology Team at Natick.

The second is data on the variability of organ size and weight. Clinical descriptions instruct physicians where to locate various organs under the body surface. These descriptions relate organs to bony features, such as rib numbers, which locate the upper and lower range for where the heart "projects" onto the front of the chest.

The surface features and landmarks necessary to register the internal anatomy can best come from the high-resolution laser digitized subjects which will be recorded and analyzed on Cyberware whole body scanners, such as the one located with the Anthropology Team at Natick.

1.2.6.1 Develop Anthropometric Modeling Software

We adapted emerging software, such as the USAF Multivariate Accommodation Model and the equivalent US Army work, and develop new software, which will has the capability to accurately generate anthropometric models accurate and appropriate for the service personnel being modeled by Virtual Prototyping. The output will be in a format compatible with the 3D Human Model Generator. Appropriate validation studies will also be performed.

1.2.6.2 Develop A 3D Human Model Framework Generator.

We developed a computer program to input the anthropometric model and output a 3D Human Model Framework. The framework consisted of body segment geometries with surfaces and joints, which conformed to the anthropometric model, and computer inertial properties. Available software and 3D databases were used where appropriate. The software has a graphical as well as a tabular output. The output will is in a format and structure so that the Internal Structure Model could be developed around it.

1.2.6.3 Develop A 3D Internal Structure Modeler.

A computer program was developed to input the 3D Human Model Framework and output an MRCMan data set which models internal human structures (tissues and organs). The modeled structures have locations, size, and mass appropriate to the size and shape of the body segment from the Framework.

Task 2 – Incorporate mass properties of human body segments and joint equations of motions for kinematic trauma and human body dynamics calculations.

Mass properties of body segments and joint equations of motion was incorporated in the Virtual Human by extracting relevant subroutines from the Air Force's *Articulated Total Body* Model (ATBM) and embedding them into the digital human. This enabled a description of the Virtual Human's response to blast, impulse, and abrupt acceleration and deceleration.

Task 3 – Development of enhanced penetrating wound and blunt trauma models

Models of penetrating wounds from ballistic impact, blunt trauma from non-penetrating projectiles, and kinematic trauma were incorporated into the digital human model. Trauma models of penetrating wounds developed by Mission Research Corporation (MRC) in its *Simulation and Assessment of Musculoskeletal Trauma from Penetrating Wounds* effort funded by DARPA's Advanced Biomechanical Technology program was exploited for that purpose. This program was funded as a Phase III effort from a previous Phase II SBIR funded by Natick. The resulting trauma models developed by MRC in this effort are being used in a lower extremity trauma and virtual surgery simulator.

Task 4 – Character simulator animation and collision detection

An articulating human model was developed for this effort. This model allows for articulation within the virtual environment and can simulate various interactions of the individual with the space he or she is located in. Each limb or movable part of the human model is treated as a separate 3D model within the virtual environment. A system parenting one limb to another is utilized to minimize the amount of data that must be communicated between modules. In this way, parenting arms and legs to a torso for example, only the x, y, z translation and rotation coordinates of the body center of mass, x, y, z, coordinates for the axis of rotation for each limb, and the angles of limb rotation about that axis will be required. This approach entails a much smaller communication bandwidth and data stream than by specifying a complete geometric database for each time step that the object changes position. In this case the bandwidth of the communication data stream is much lower.

The geometry is made of triangular polygons with simple materials and textures. Sufficient polygonal density exists to ensure adequate representation of the digital phantom. The data files describing the digital phantom contain all data structures required by the visualization tool (e.g., Coryphaeus®). Three dimensional coordinates for each vertex, vertex connectivity table for each polygon, axis of rotation and rotation constraints for each movable element, and material, and texture definitions. These files are created during the Session Configuration phase of the simulation.

Injuries or limitations in range of motion due to protective equipment are simulated by imposing rotation constraints on the affected joint or axis of rotation. These "limitations" are computed by one of the Analysis Modules. For example, if the "player" is shot in the left shoulder, and the damage to the shoulder (calculated by *JOHN O*. – MRCMan) would result in a disabled left arm, the "player" cannot use the simulated left arm or hand to manipulate any virtual device or weapon.

Task 5 – Virtual Equipment Models

Existing models of protective equipment performance (Natick/MRC contracts DAAK60-92-C-0008 and DAAK60-91-C-0087) were extended to describe the interaction of military projectiles with multilayer soft fabric body armor, with and without rigid inserts.

Task 6 – Interface development

A general-purpose application programming interface (API) for software/visual models of protective equipment and the assignment/attachment of these models to characters in the battlefield simulation were developed.

Task 7 – HLA Compliance for Soldier Character Simulator

High Level Architecture (HLA) has been designated as the standard technical architecture for all DoD simulations by the Under Secretary of Defense for Acquisition and Technology. Since compliance with HLA will be required, this effort will follow the requirements of HLA. These

requirements direct that each simulation (i.e., a "federate" in HLA terms) and each group of interoperating simulations (i.e., a "federation") follow a set of HLA Rules. They also require development and documentation of HLA Simulation Object Models (SOMs) and an HLA Federation Object Model (FOM). Interfaces were developed to the HLA Runtime Infrastructure (RTI) software.

A major purpose of the HLA is to provide a common architecture that allows multiple simulations to interoperate without the need to develop specific interface software for each combination of simulations. This promotes simulation/software reuse with minimal effort.

HLA uses the term "federation" to represent the cooperative set of simulations, display systems, loggers, etc., that interoperates for a particular purpose. The HLA refers to each member of a federation as a "federate". Federates cooperate to model entity states and interactions for a particulate domain of interest; an entity in HLA is generically referred to as an "object". An object's state is represented by the values of its "attributes". "Interactions" between federates may cause the attributes of objects to changed. HLA supports the concept of shared and dynamic ownership for objects attributes across a federation, i.e., different attributes of the same object may be owned by different federates. However, each attribute of each object may be owned by only one federate at a time. Attribute ownership implies the ability and responsibility to update that attribute's value for that object. It also implies the responsibility to "publish" changes to attribute values for other federates that have "subscribed" to that attribute.

A federate can be an active participant in the federation, i.e., the federate owns objects, publishes attribute changes, optionally subscribes to attributes of objects in other federations, and optionally generates or responds to interactions. A federate could also be a passive member of a federation, i.e., the federate only subscribes to the attributes objects owned by other federates.

The HLA rules require the development of a Simulation Object Model (SOM) for each federate participating in a federation and Federation Object Model (FOM) for the federation as a whole. The SOM and FOM must be documented using the HLA Object Model Template (OMT). The use of the OMT promotes the long term and wide usability of a federate by requiring a common descriptive format for the public interactions of that federate. The OMT requires the definition of the relevant information about public object classes, their attributes, and interactions supported by the federate. Support can be in the form of publish, subscribe, or both. The SOM defines the public interactions supported by an individual federate, and can be used to support multiple federations. The FOM defines the interactions between a particular set of federates cooperating to represent a domain of interest.

HLA provides common software to support the interaction of federates in a federation. This software, the HLA Runtime Architecture (RTI), provides a set services required to execute the HLA concept. There are six basic RTI service groups: Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management. Each of these services is accessed through an RTI Application Programmer's Interface (API), that defines the data types, structures, and functions required to interact with the RTI. The F.0 version of the RTI for the Sun Solaris environment using a C++

API is scheduled to by available in December 1996. This version supports most of the functionality required in the HLA 1.0 Interface Specification. Additional versions for other UNIX environments, including the SGI, are scheduled to be available in early 1997. Other versions supporting the Windows/NT environment are likely to be available later in the year. This effort will require the development of interfaces to the HLA RTI using the API.

Task 8 – Design and analysis module development

MRCMan will be immersed into a real time simulation environment to complete the implementation of the virtual prototyping system. One such simulation environment commonly in use in DoD activities is Coryphaeus®. MRC is using this approach in Development of its *Urban Warfare Virtual Environment* for STRICOM. In fact in exchange for giving Coryphaeus first right of refusal for commercial rights to the resulting software, Coryphaeus has given royalty-free use of it s software for the duration of the development effort.

In the proposed development effort, a hybrid of *Commercial Off-The Shelf* (COTS) and *wrap-around* custom software will be employed. Coryphaeus® is presently the leading candidate for providing the core visualization and real time simulation capabilities. A custom software development effort will be utilized to create the desired capabilities which are not available in Coryphaeus®. *JOHN O.* (MRCMan) will be modified to export all anatomical parameters to the real time simulation portion of the virtual prototyping system. *JOHN O.* (MRCMan) will in a sense become the compute server for the entire system.

Several geometric databases will be required for the personal protective equipment virtual prototyping system. The key databases are:

- Human models (various sex, race, and anthropometric categories)
- Personal protective equipment
- Workspace
- Weapons and other equipment
- Environments
- Facial identification and clothing
- Wounds

A user interface will be developed to configure and monitor the virtual prototyping simulation. It will have areas for selecting all items listed above. Since it would be prohibitive to create 3D models for all possible combinations of digital human models, only the most common combinations would be immediately available at runtime. Any “non-standard” digital human will be configured with the interface and information sent to the *JOHN O.* (MRCMan) module to build the required geometry. This polygonal geometry will then be loaded into memory for the current session and will then also be added to the list of available human models for future use.

Three dimensional models of the personal protective equipment could eventually be created either in a CAD package or within the Coryphaeus® modeling environment. If the protective equipment is still under development, the designer can export a model from the CAD tool he/she

is using and load it into *Designer's Workbench* for final preparation. Again, once these models have been created the first time they will become part of the library and will be available for future sessions.

A library of vehicles, weapons, and environments could be created much in the same way. Also, since many of the desired vehicles, weapons, and environments have already been created for other DoD and commercial uses, they may only require converting to a new format to be made available for use in a virtual prototyping session.

When two or more characters are immersed, unique identification will be crucial for effective simulations. Thus, a library of "faces" and clothing will be created. The faces will be simple texture maps created from color photographs. Front and side photos will be digitally "stitched" together to create a texture which can be applied to any digital human for unique identification. Various materials and colors will be available for clothing.

The digital human was articulated using the *Motivate*™ software package. This model can be articulated within the virtual environment and is able to simulate the interaction of the individual and the space or vehicle he or she is located in. The model can also reflect changes in range of motion due to protective equipment the individual is wearing or due to wounds. There are three aspects to this fully articulating human model:

- 3D geometry, materials, and textures
- XYZ coordinates and rotation constraints for all joints, limbs, and movable parts
- Navigation and manipulation input and feedback links

The geometry is made of triangular polygons with simple materials and textures. Sufficient polygonal density exists to ensure adequate representation of the digital phantom. As discussed, the geometry will be created based on sex, race, and anthropometric specifications, selected from a pre-defined list, or acquired from the Natick Cyberware body scanner. Texture maps will be used to add specific faces for individual identification.

The data files describing the digital phantom will contain all data structures required by Coryphaeus®. Three dimensional coordinates for each vertex, vertex connectivity table for each polygon, axis of rotation and rotation constraints for each movable element, and material, and texture definitions. These files are created during the setup phase of the simulation.

There are several methods which can be implemented to determine the functionality of a particular piece of protective equipment with respect to the work space. Pre-programmed operations and movements within a particular workspace combined with collision detection routines to identify interference between components is one option. This option would not require real time visuals but instead could utilize post-processed visualization.

Another option, is to have multiple "players" immersed in a VR environment. Each player would wear a motion suit or other positional feedback system to detect displacements limbs and joints. This type of "motion suit" is common in real time motion capture applications. These players

would also wear the virtual personal protection equipment and would virtually interact with the workspace. Any lack of mobility would show up as an inability to perform various operations.

There are also several solutions between the multiple player *man-in-the-loop* immersive environment and the non-immersive single player environment. For example, instead of having completely immersed players, we could have one player immersed and interacting with a workspace. In this way, only the parts of the player which the player can see need be visually represented; arms, hands, legs, and feet. This simplifies the visuals and the VR hardware significantly.

If the "player" sustains some type of injury, the limitation in range of motion or limited use of a limb can easily be integrated into the simulation by limiting the effectiveness of the VR hardware. For example, if the "player" is shot in the left shoulder, and the damage to the shoulder (calculated by *JOHN O.* – MRCMan) would result in a disabled left arm, the "player" would not be able to use the simulated left arm or hand to manipulate any VR device or weapon.

The design of protective equipment as conceived of in this effort has three basic cycles: static design, dynamic design, and field testing. In each of these phases the application of 3D visualization can be used to improve the design process.

Static Analysis

During static analysis, various factors are evaluated and engineering trade-offs are made. The design of personal protective equipment must balance factors of wearability, durability, cost, and overall effectiveness of protection. Successful completion of the proposed Phase II effort will provide a significant increase in the amount and quality of the data available to the designer. To make effective use of this influx of data, the designer needs improved tools to evaluate the results of trade-offs made during each design iteration. Using 3D modeling and rendering provides the foundation for such a tool set.

The overall effectiveness of personal protective equipment is a balance of the greater encumbrance imposed by the equipment versus the higher probability of mission completion through reduced casualties. Here again, 3D imagery can be used to assist in the design process. This is accomplished by identifying the range of threat environments in which the equipment is to be used, and the distribution of injury severity and type which can be expected in those environments. This is then translated into a visual model indicating areas of the body at greatest risk to different types of injury (i.e., penetrating wounds, blunt trauma, and heat stress). The designer can then interactively apply protective material, in a selective manner, until an acceptable risk level is reached. The designer can thus use a minimum of mass, thereby reducing encumbrance, while still providing the maximum level of protection at a given cost point.

Wearability is a function of several factors such as mass distribution, abrasion, heat transfer, and flexibility . The 3D image can present the designer with color coded representations of how the proposed equipment interacts with a representative selection of wearers. Given the current and short term analysis capability, the main factors to be evaluated at this stage are protection, range

of motion, mass distribution, total weight, and heat transfer. As analytical models of human mobility and range of motion improve, these models can be incorporated into this phase of the analysis as well.

Combining analytical models of mobility and range of motion with material property data will allow evaluation of protective equipment for durability and potential encumbrances to range of motion. Improved durability can be obtained by identification, as early in the design cycle as possible, of points of high stress or wear. This information permits the designer to select materials and construction techniques with greater value and performance at a lower cost. The implementation of the system will have major components as shown in Figure 1-7.

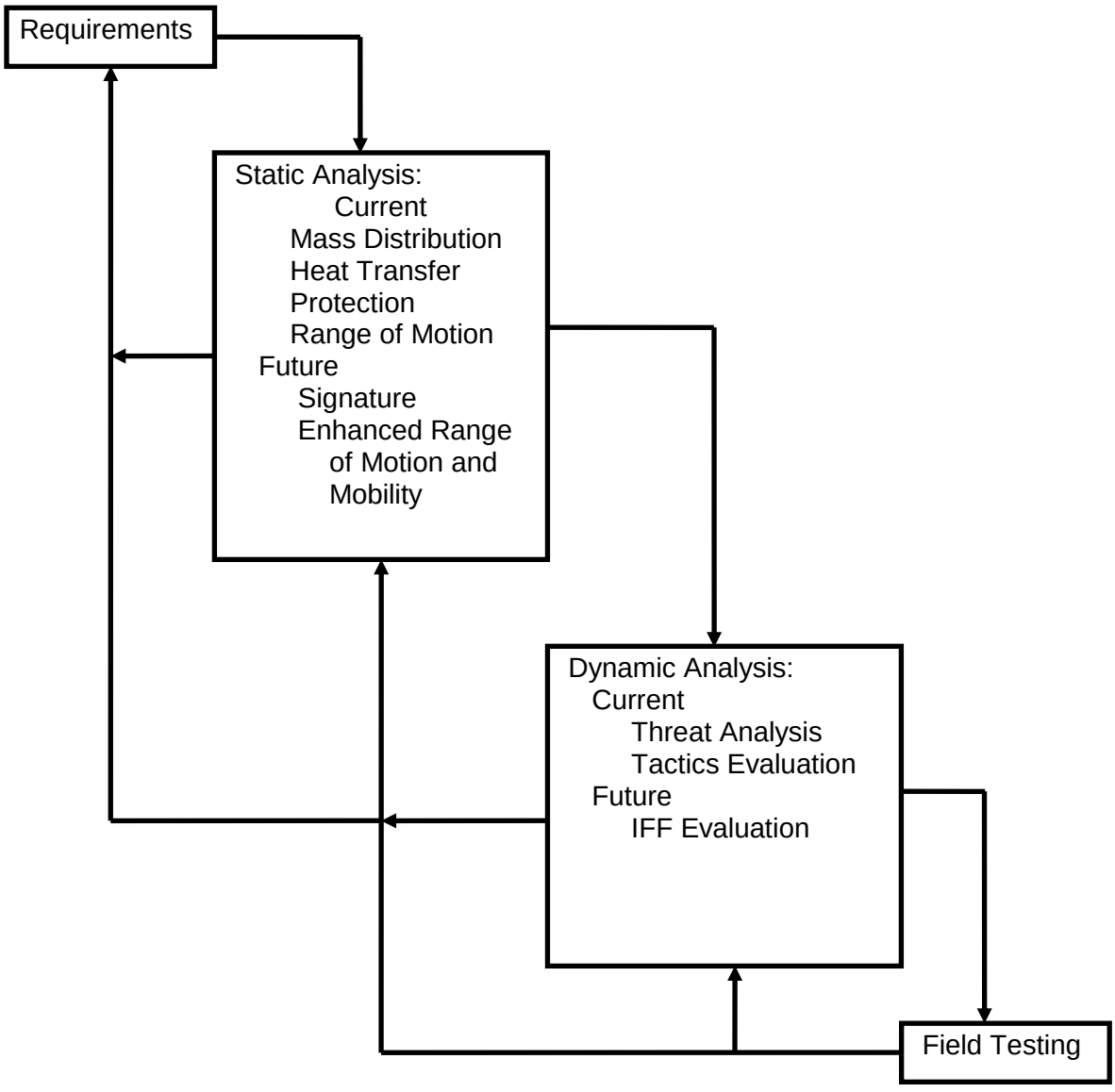

Figure 1-7. Relationship of Various Analysis Options in Virtual Prototyping System

During the initial design phase, the designer reviews the effect of the projected threat environment. Given a representative selection of unprotected anthropomorphic models, the designer can create a color coded visualization of the projected distribution and severity of different classes of injury. This process can be repeated with the models protected by current issue equipment. As the design matures, the new equipment design is evaluated with respect to the same criteria as applied to the unprotected and currently protected model. The designer can then be presented with a side by side comparison of the effectiveness of the proposed design.

Since the evaluation process is based on a detailed anthropomorphic model, the designer has the option of examining detailed representations of the characteristics of selected wounds. For example, if the general evaluation of the protective equipment to a certain group of anthropomorphic models indicates a high occurrence of trauma to the lower back, the designer can move into the body and examine the nature and extent of the trauma. This form of evaluation can provide the designer with insight needed for innovative solutions (see Figure 1-8).

Dynamic Analysis

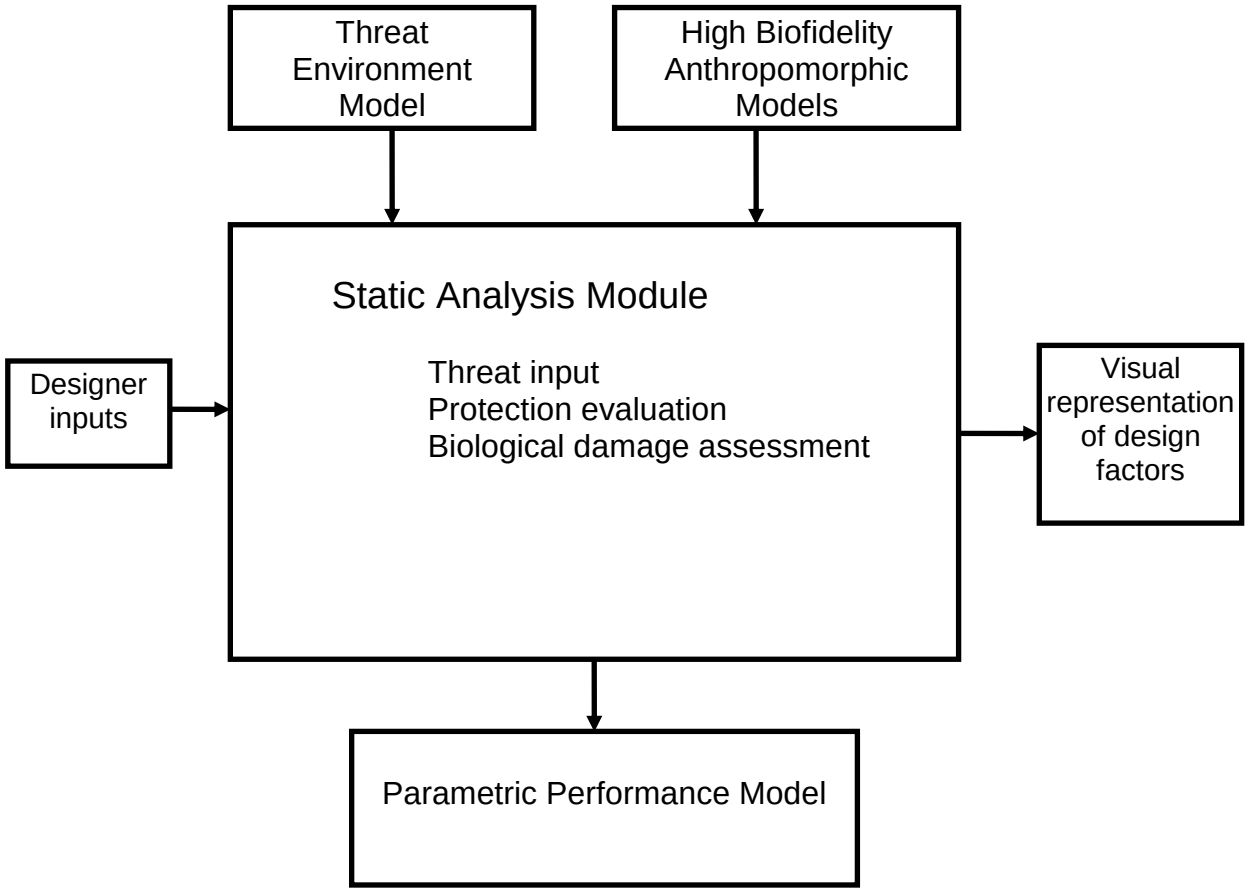

Figure 1-8. Flow of Information in Static Analysis Module

Once a cycle of static design is competed, a dynamic analysis is needed to validate the conclusions of the static analysis (Figure 1-9).

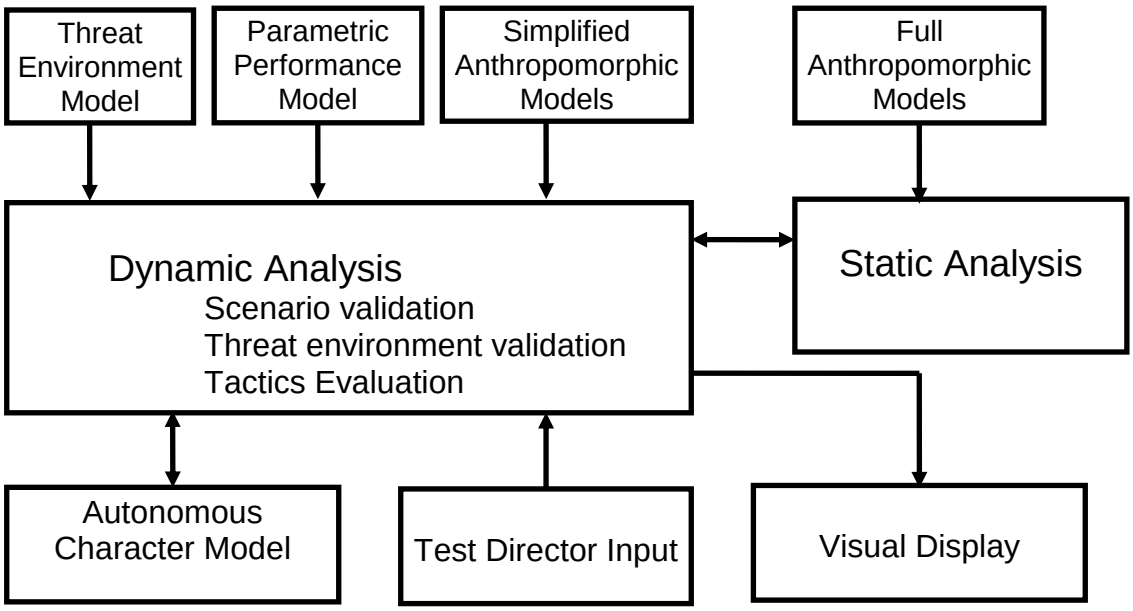

Figure 1-9. Flow of Information in Dynamic Analysis Module

The dynamic analysis looks for such features as significant changes in bulk which would effect the selection of cover or access to confined spaces. Significant changes to endurance due to better heat transfer or reduced mass, could effect the selection of tactics. The dynamic analysis would take synthetic actors/combatants through a series of scenarios. Each run would be scored against a baseline simulation. The simulation would be controlled by a combination of self-directed characters and a human director who could make use of new tactical opportunities.

The parametric model provides a level of abstraction from the high fidelity models used in the static analysis. This makes the parametric model more appropriate for use in dynamic analysis. Yet this does not preclude the use of the static analysis tools during dynamic testing. If at any time the test director wishes to validate an event in the dynamic analysis, the dynamic analysis can be suspended and the desired events run through static analysis to obtain more detailed results.

The static analysis provides the designer with statistical data on the type and distribution of impacts from which the equipment is suppose to protect the wearer. The dynamic analysis will demonstrate that the original threat model is consistent with the manner in which the equipment

is used. The results of these tests would be passed back to the static analysis to provide a more complete analysis of the protective equipment.

When the design is approaching the point where prototype equipment is to be built, a virtual environment can be used as a final human factors check of the protective equipment. The Urban Warfare Virtual Environment being developed by MRC for will be compatible with the Natick virtual prototyping system and could be exploited as the intermediate step between theoretical analysis and actual field testing alluded to earlier.

However, the response of teams using the new equipment can also be evaluated. Given a threat environment, a team using baseline equipment would normally select one of several tactical solutions to resolve the situation. Since no solution to a combat situation is risk free, every solution must be evaluated on a statistical basis, weighing team losses against mission success. By substituting the proposed protective equipment, new tactical options can be evaluated as well as changes to loss rates using conventional tactics.

Once development of the virtual environment and character simulator are sufficiently mature, follow-on activities could include, for example, evaluating visibility and signature issues. Do team members find it easier or harder to identify other team members? What is the perception of equipped team members from the point of view of the aggressors?

An example of this issue would be the effect of a characteristic Infra Red pattern resulting from the heat transfer characteristics of the protective equipment. If only the team members have suitable IR equipment to detect this pattern, the pattern becomes an aid to recognition and reduces losses to friendly fire. However, if the aggressor has the ability to recognize this pattern,

it only makes team members a target, thereby reducing the overall effectiveness of the team. By running a number of simulations in a virtual environment, the evaluators can quickly develop an appreciation of the new equipment's strength and limitations even before the prototypes have been fabricated. By the time actual field testing begins, the group doing the evaluation has had an opportunity to develop a more comprehensive set of tactics to apply. This should result in shorter, more effective field testing.

System Overview

The complete system as represented schematically in Figure 1-10, is made up of a development environment and multiple functional modules. The development environment operates independently of the functional modules. The functional modules are integrated together to form a robust and modular software environment applicable to many types of simulations.

Development Environment

The purpose of the development environment is to enable the creation of new 3D models (weapons, equipment, tools, etc.) and virtual "worlds" (3 story building, subway, parking garage) for use in real time simulations. Some models such as the personal protective equipment or the

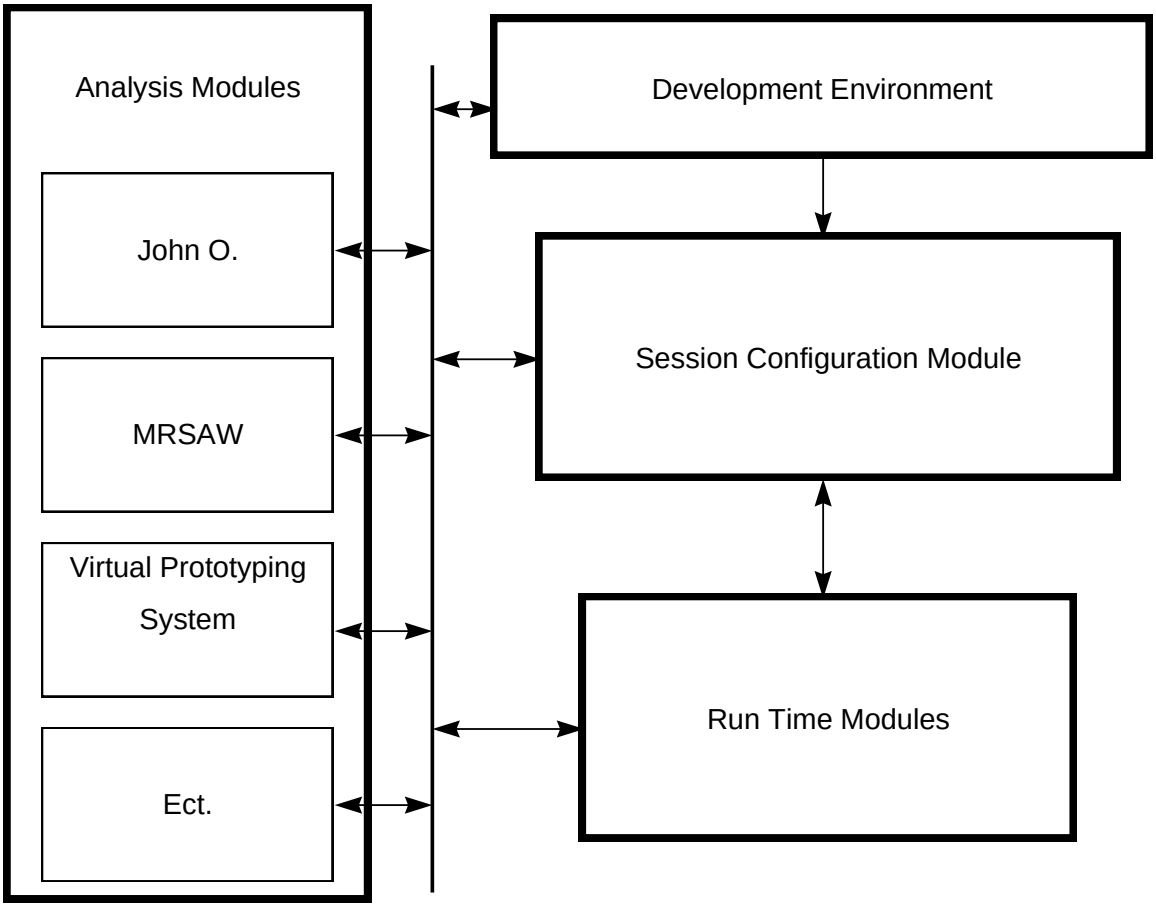

Figure 1-10. Overview of System

human body models will be created in one of the Analysis Modules but others may require some final processing in a 3D modeling tool specifically designed for real time use. A tool which specializes in creating real time simulations will be required for development of the specific virtual environment used for "testing" the various prototype designs.

One such set of modeling and real time simulation software tools commonly in use in DoD activities is Coryphaeus®. The combination of Designer's Workbench® and EasyScene® provide the core capabilities for developing models and environments for use in real time simulation applications.

An environment consisting of a couple rooms separated by a large open space with a few various walls and doorways, for example, would be created in Designer's Workbench. Other objects (human model, basic personal equipment, workspace) would be created in the Analysis Modules incorporated into the prototyping system and exported to the proper format for use in the virtual environment. Elements of the simulation would then be selected and "assembled" into the

environment model via the Session Configuration Module described below. The Session Configuration Module "compiles" everything together to create a complete Run Time Module

Session Configuration Module. Ideally, a graphical user interface should be developed to manage the selection, configuration, and monitoring of all geometric data, analysis data, CPU processes, and computer hardware necessary for the virtual prototyping simulation. This GUI would essentially run as a shell on top of Coryphaeus EasyScene. When all parameters are properly selected by the user, an "execute" feature will assemble all information into an EasyScene based run time module. This process is analogous to a pre-processor which creates lines of "C" code for a specific purpose and then compiles the code into an executable form of the "C" program. In this example, the executable "C" program is a run time module.

The Session Configuration Module will allow the user to select the following types of information:

- Human models
- Personal protective equipment
- Weapons and other equipment
- Virtual Environments
- Analysis Methods
- Graphics Hardware being used
- Other available network resources

From the standpoint of human models, since it would be prohibitive to create 3D models for all possible combinations of digital human models, only the most common combinations would be immediately available at runtime. Any "non-standard" digital human will be configured with the graphical interface and information sent to the *JOHN O*. (MRCMan) module to build the required geometry. This polygonal geometry will then be loaded into memory for the current session and will then also be added to the list of available human models for future use. More exact and detailed human models could be created using the Natick whole body laser scanner. This data would also be processed by *JOHN O.* (MRCMan) which would export the model in the polygon format required by the real time software.

Three dimensional models of the personal protective equipment will be created either in a CAD package, within the Coryphaeus® modeling environment, or scanned with the whole body laser scanner. Again, once these models have been created the first time they will become part of the library and will be available for future sessions.

A library of vehicles, weapons, and environments could be created much in the same way. Also, since many of the desired vehicles, weapons, and environments have already been created for other DoD and commercial uses, they may only require converting to a new format to be made available for use in a virtual prototyping session.

Depending on which analysis information has been requested by the user via the GUI, specific information will be sent to the appropriate analysis module. The analysis module will in some

way affect the data which was sent to it and return new values. The analysis modules are discussed in more detail below.

The ability to identify the graphics hardware being used allows a wider range of users to execute the virtual prototyping software. For example, users who have access to high end SGI Onyx2 level graphics would benefit by receiving higher fidelity models from all the analysis modules and a greater number of "immersed" objects. On the other hand, users on low end graphics hardware would have fewer and lower resolution objects available for the simulation.

From the standpoint of network resources, if the GUI "is told" of the availability of additional processors, it will allow more analysis to be performed as necessary during the simulation. This could also result in higher fidelity graphics quality.

Analysis Modules. The analysis modules shown in Figure 1-10 are a series of stand alone processes which calculate a specific parameter as required for the real time simulation. For example, the *JOHN O.* (MRCMan) module combines anthropometry, internal geometries, and dynamic properties to create a 3D human model. The FATIGUE analysis module (and/or IUSS interface), combines 3D human model data, protective system data, and environmental factors to create heat stress distribution for the human model. Still another module, MRSAW (Mission Research Small Arms Weapons) uses weapon and munitions data and 3D environment geometry to predict the trajectory of bullets and munitions fired in the virtual environment. Additional analysis modules can be developed and integrated into the overall system at any time with little change to the overall system architecture.

There are two main types of analysis modules; those which process static data and feed results into the simulation during the session configuration phase, and those which require information from the simulation as it progresses to update calculations.

The operation of each analysis module can be managed with the session configuration module discussed above or with a dedicated interface. Actually, the dedicated interfaces for each module are what make up the "analysis module manager" section of the session configuration module so it will be a relatively straight process to isolate them for standalone use.

Run Time Modules. A run time module (Figure 1-10) is a "compiled" version of all elements necessary for the simulation. Pointers to all geometric data sets, analysis data sets, analysis modules required for real time analysis, and everything else specified with the session configuration module are stored in these files. File formats are a function of the real time software constraints (e.g., Coryphaeus) but typically binary and ASCII files are used.

These run time module files can be re-opened with the session configuration manager to change one or more of the parameters (human model, personal protective equipment, etc.) or they may simply be executed again for a new interactive session. In the new interactive session, the entire 3D environment will be identical to the previous session but any input from external devices (mouse, joystick, etc.) will cause the outcome of the simulation to be different.

During the session, an option can be made available to “record” the entire simulation to disk for playback at a later time. This “playback” file can grow quite large depending on the complexity and length of the simulation. Playback files will be useful as a means of comparing with other similar sessions of the simulation. For example, one could investigate the effect of different combat tactics by keeping everything else constant between sessions. Then by using the session configuration module to display side by side windows, one could view the simulation for various tactics.

Management of all runtime modules files will be accomplished using the session configuration module.

2. ANATOMICAL AND ANTHROPOMETRIC MODELING

2.1 OBJECTIVES

2.1.1 Update the 3D anatomical structure data for MRCMan using the Visible Human (VH) data. The objective was to produce anatomical data sets in the MRCMan format, but using the VH data indexed to 1600 structures by Gold Standard Multimedia (GSM). The reason for the change was to increase the level of detail in the anatomical model, and also to have sufficient data to produce data sets with different anthropometric dimensions.

2.1.2 Develop the capability to modify the size and shape of human anatomical data to conform to different anthropometric profiles. The objective was to be able generate automatically and easily new MRCMan data sets representing the size and shape variations found in current U. S. soldier populations. The reason for this was to better validate the testing of prototype protective equipment in the MRCMan model.

2.1.3 Generate anthropometric profiles representing the U. S. Army population. The objective was to use the current Army database, the 1988 U. S. Army ANthropometric SURvey (ANSUR), to compute anthropometric dimensions to drive the generation of new anatomical data sets. The reason was that the virtual prototyping was primarily for Army equipment and therefore would be more valid if modeled on Army soldier data sets.

2.1.4 Generate Articulated Total Body Model (ATBM) input data. The objective was to modify a previous program, GEBOD, to produce input data sets for the ATBM using the ANSUR database. The reason for this was to have ATBM and MRCMan data modeling the same individuals.

2.1.5 Develop the capability to fit remodeled VH data to 3D human surface scan data sets. For MRCMan, the anatomical data was required to fit a set of linear anthropometric dimensions (lengths and widths of body segments). The objective in this task was to fit the VH data to specific individuals represented by high resolution 3D surface scan data sets.

2.2 VISIBLE HUMAN DATA

2.2.1 The original and GSM-edited VH data set. The Visible Human Project is sponsored by the NIH National Library of Medicine. To date, two humans have been processed and very high resolution anatomical data has been released for the use of scientists and industry (see Appendix 1 – Visible Human Project Fact Sheet). The Virtual Prototyping project used data derived from the first person processed, a male. The subject was a Texas prisoner, executed, then flash-frozen. The subject was not an average person in size or shape, being much more heavily muscled (Figure 2-1). This is especially apparent in the neck, where the development of muscles is far larger than normal. The subject was positioned so that his arms were close by his side, forearms pronated, and hands resting on top of his thighs. This position was apparently necessitated by his

FIGURE 2-1. The Visible Human Male

large size and the need to position him in CT scan and MRI machines. The consequences of this positioning were to make our editing of the data considerably more difficult.

After the body was scanned using CT and MRI, it was sliced at 1-mm increments and the exposed surfaces digitally photographed (Figure 2-2). Gold Standard Multimedia used these images, at 0.33-mm pixel resolution, to index the anatomical structures throughout the body. The result is a database of 1871 sections, each section recorded as an indexed point in a 2D array. This database was purchased/licensed by MRC for use in this project.

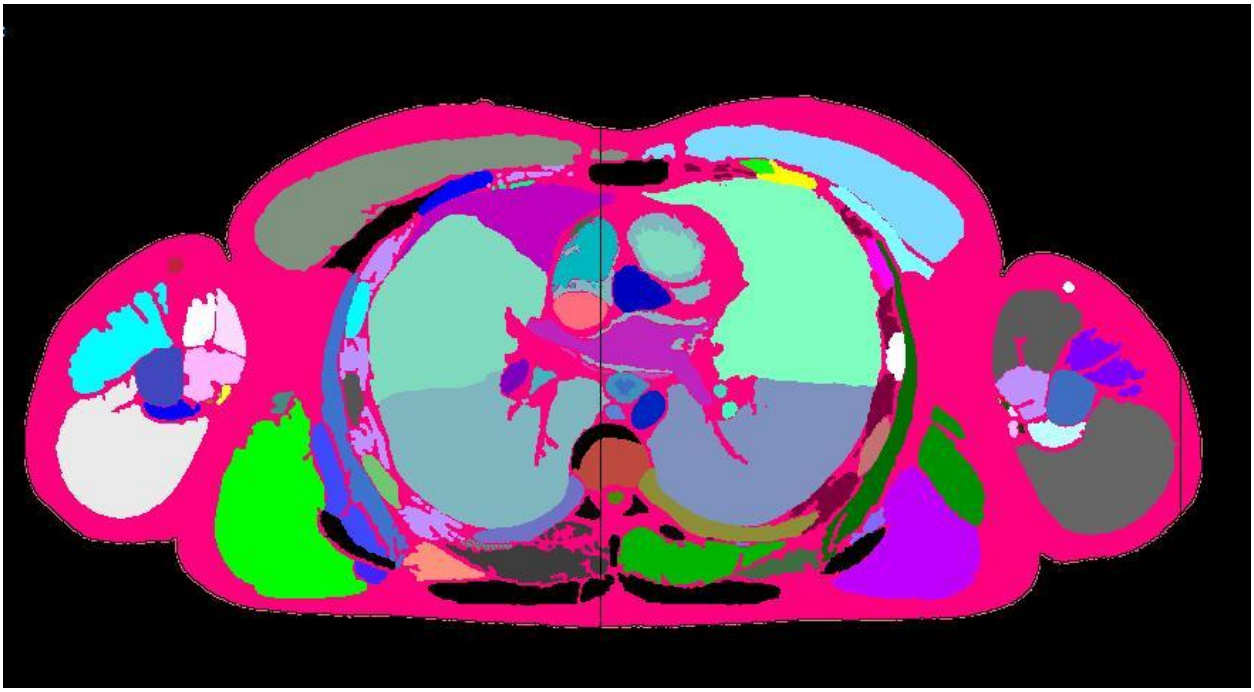

FIGURE 2-2. A slice from the GSM indexed data base.

2.2.2 Editing GSM data into body segments. The MRCMan anatomical model is divided into fourteen segments: Head/Neck, Thorax, Upper Arms, Forearms, Hands, Thighs, Calves, and Feet. The format for the cross-sectional data is with the Head/Neck divided into slices 12mm apart vertically, with each slice consisting of data points in a 5-mm resolution grid. Other body segments have slices 25-mm apart. Data for each body segment is stored as a separate file.

We modified an existing program, ShapeAnalysis, to edit the GSM data into body segments. Because of the amount of data involved, the lesser resolution needed for MRCMan anatomical modeling, and the time available, only the even-numbered slices were edited. Each GSM slice was input, and where there was data for more than one body segment, the borders between segments were marked graphically, and an output function wrote the data as separate files marked for the appropriate segment. No data was lost in this process – the reformatted data files retain the 0.33mm resolution for indexed anatomical structures. The result of this process is a directory of 1867 files occupying approximately 1.8 GB of disk space.

In addition, anthropometric measurements and landmarks needed for the Internal Structure Modeler were also taken from the VH data slices. The measurements provided lengths, widths, and depths for each segment. Landmarks specified local axes which were then used to displace the VH segments into axes similar to those used in other 3D body analyses. In general, the axes orient the body segment such that the z-axis is parallel to the long axis, and the x- and y-axes are oriented mediolateral and anterior-posterior.

2.3 GEBOD

Program *GEBOD* (GENerator of BODy data)is an interactive DOS program which automates production of body description data used as input for the *Articulated Total Body Model* (ATBM). The ATBM is a dynamic rigid body model developed by the US Department of Transportation and maintained by the Air Force. It simulates the motion of the human body during dynamic events such as automobile crashes and aircraft cockpit ejections. GEBOD provides such information as mass, center of gravity location, contact surface dimensions, principal moments of inertia and their associated directions for a minimum of 15 body segments, which are modeled as ellipsoids. It also provides the location, orientation, and characteristics of 14 body joints. Data can be obtained for several subject types: 1) human child, 2) adult human Air Force female, 3) adult human Air Force male, 4) Hybrid II manikin, 5) Standing Hybrid III manikin, and 6) Sitting Hybrid III manikin. The option to obtain data for adult human Army female and adult human Army male subject types has recently been added for the Visual Human project.

When selecting the adult human Air Force subject types the program allows the user to choose whether he or she is interested in outputting data for 15 body segments or 17 body segments. The 15 body segments include: head, neck, thorax, abdomen, pelvis, right and left upper arm, right and left lower arm plus hand, right and left thigh, right and left calf, right and left foot. The 17segment option is different in that the right and left lower arm and hand segments are separate rather than combined. The adult human Army subject types automatically output data for 17 body segments.

The basic principal behind GEBOD is the use of regression equations. The statistical analysis of existing human body stereophotogrammetric data sets (Young et al., 1983, McConville et al., 1980) was used to develop regression equations to predict segment mass properties and three-dimensional joint locations. Regression equations for adult human anthropometric dimensions are based on the 1967 Air Force male survey (Grunhofer and Kroh, 1975), the 1968 Air Force female survey (Clauser et al., 1972), and the 1988 Army survey of males and females (ANSUR, Gordon et al., 1989).

The equations for the adult human options are based on stature and/or weight. The child option can also use age as a predictor. The user is simply asked to select the predicting dimensions to use and input the value of the dimensions. These predicting values are input into the regression equations, which compute most of the information described above. At least 31anthropometric dimensions are predicted and are used to compute the values of the remainder of the data. It is

possible for a user to input a file containing the values for the entire set of 31anthropometric dimensions, if desired. The anthropometric dimensions are shown in Table 2-1.

TABLE 2-1. Anthropometric Dimensions

1. Weight	2. Standing Height
3. Shoulder Height	4. Armpit Height
5. Waist Height	6. Seated Height
7. Head Length	8. Head Breadth
9. Head to Chin Height	10. Neck Circumference
11. Shoulder Breadth	12. Chest Depth
13. Chest Breadth	14. Waist Depth
15. Waist Breadth	16. Buttock Depth
17. Hip Breadth, Standing	18. Shoulder to Elbow Length
19. Biceps Circumference	20. Elbow Circumference
21. Forearm Circumference	22. Wrist Circumference
23. Knee Height, Seated	24. Thigh Circumference
25. Upper Leg Circumference	26. Knee Circumference
27. Calf Circumference	28. Ankle Circumference
29. Ankle Height, Outside	30. Foot Breadth
31. Foot Length	

In addition to the basic set of 31 dimensions, output for 17 segments includes dimensions describing the hands (breadth, length, and depth). Furthermore, the adult human Army subject type options output three more anthropometric dimensions useful specifically for the Visual Human project: Crotch Height, Trochanterion Height, and Cervicale Height.

In simulating human body motion, it is important to have a realistic model of the human body, itself. The methods used by GEBOD, based on actual data, provide this model and are highly adaptable to creating input for diverse motion simulations.

GEBOD produces two output files: One with the format for input to the ATBM (*.ain), and another with a format for input to the Internal Structure Modeler (*.tab).

2.4 INTERNAL STRUCTURE MODELING SOFTWARE DEVELOPMENT

2.4.1 Programming objectives. The objective was to develop software that could input anthropometric profiles from GEBOD, and then compute from the VH data a new set of body segment slices that conforms to the GEBOD anthropometry and MRCMan file formats.

2.4.2 Program description. The program executable is ism.exe and was developed in C using Microsoft Visual C++.

In the MRCMan body model, each body segment is oriented parallel to the segment long axis. Because of the position of the VH body when it was sectioned, the segments in those data are not oriented correctly. The ISM program creates a new set of segmented body slices oriented to the long axis of each segment, and with point distances and slice separation distances set by the user. Thus, the VH data is re-sliced, or re-sampled, to create a new data set from the original points. The program works by taking VH data, slice by slice, displacing the slice so that it is in a segment local axis system with a z-axis parallel to the long axis of the segment, then searching the points in the slice to see if any are close to the grid of the cross-section being constructed.

The program is initialized by the input of an anthropometric file representing the subject (target in MRCMan format) to be modeled. After the user specifies the distance between slices and the resolution of the cross-sectional points, the program performs the same operations on each body segment.

1. Anthropometric variables are used to compute the needed MRCMan segment length, width, and depth measurements.
2. The MRCMan measurements are compared with comparable VH measurements to compute scaling parameters.
3. The segment length parameter is used to compute the distance between slices of the VH data that will result in the MRCMan length at specified slice distances.
4. The width and depth parameters are used to compute scale factors to be applied to each VH slice before re-sampling.
5. Beginning at the top of the body segment, each VH slice is input, then displaced to the local axes. Using the scaling parameters, the slice is then rescaled in the local axes X-Y plane to conform with the MRCMan width/depth measurements.
6. The VH slice is then searched to see if any points are within +/- ½ length increments of the target slice plane. That is, if the target slice is to be at a Z-level of 100.0, the distance between slices is 25.0mm, and the VH slice (after displacement) has any points with Z-coordinates between 87.5 and 112.5, then the VH slice qualifies for resampling.
7. If the VH slice qualifies by being close to the sampling plane, the VH slice is searched to find points within +/- ½ cross-sectional increments (the grid points on the sampling plane). If they exist, and are closer to the MRCMan cross-sectional (X-Y) coordinates than previous VH slice points, the point and its anatomical index is stored for possible inclusion in the final target slice.
8. After all qualifying VH slices are searched, the resulting MRCMan slice is written out in MRCMan format.
9. The Z-level is decremented by the distance to the next desired MRCMan slice, and the process of searching and sampling the VH slices is repeated.
10. After all target slices are completed for a segment, the file is stored in MRCMan format, and the next target (MRCMan) segment is created.

2.4.3 Operation. The executable and input files should be in the same directory:
ism.exe
*.tab [anthropometry file from GEBOD]
Segment local axis files are shown in Table 2-2.

TABLE 2-2. Segment Local Axis Files

headaxis	leftarmaxis	leftfootaxis	leftforearmaxis	lefthandaxis
leftlegaxis	leftthighaxis	rightarmaxis	rightfootaxis	rightforearmaxis
righthandaxis	rightlegaxis	rightthighaxis	torsoaxis	

Fifteen files for input to MRCMan are created shown in Table 2-3.

TABLE 2-3. MRCMan Input Files

_tissue.hed	_tissue.lca	_tissue.lfa	_tissue.lft	_tissue.lhn
_tissue.lth	_tissue.lua	_tissue.rca	_tissue.rfa	_tissue.rft
_tissue.rhn	_tissue.rth	_tissue.rua	_tissue.tor	master.tis

The program is started through ism.exe. The user is then prompted for the GEBOD anthropometry file name, the distance between grid points on each output slice, and the distance between slices. As the program runs, progress is written to the DOS window, as well as being written to the file audit_track.txt.

Input:
- Anthropometry file (*.tab)
- Segment local axes files (*axis)
- Distance between points in the output cross-sectional slice (prompted)
- Distance between slices (prompted)
- Segmented VH data files

Output:
- Sampled data in MRCMan format
- MRCMan master.tis file
- audit_track.txt with run-time information

2.4.4 Validation testing. Two aspects of the results needed to be tested: (1) The functioning of the data files in the MRCMan model, and (2) the anthropometric measurement of the resampled data sets.

The output from several test runs of the ISM was used as input for current version of MRCMan. The data was input and initialized by the program without error. Mark West has written that he believes the data is valid and functioning in the tests of MRCMan that he has performed.

Three data sets representing short, medium, and tall/thin persons were computed and the resulting reshaped slices were measured for conformity with the GEBOD anthropometry:

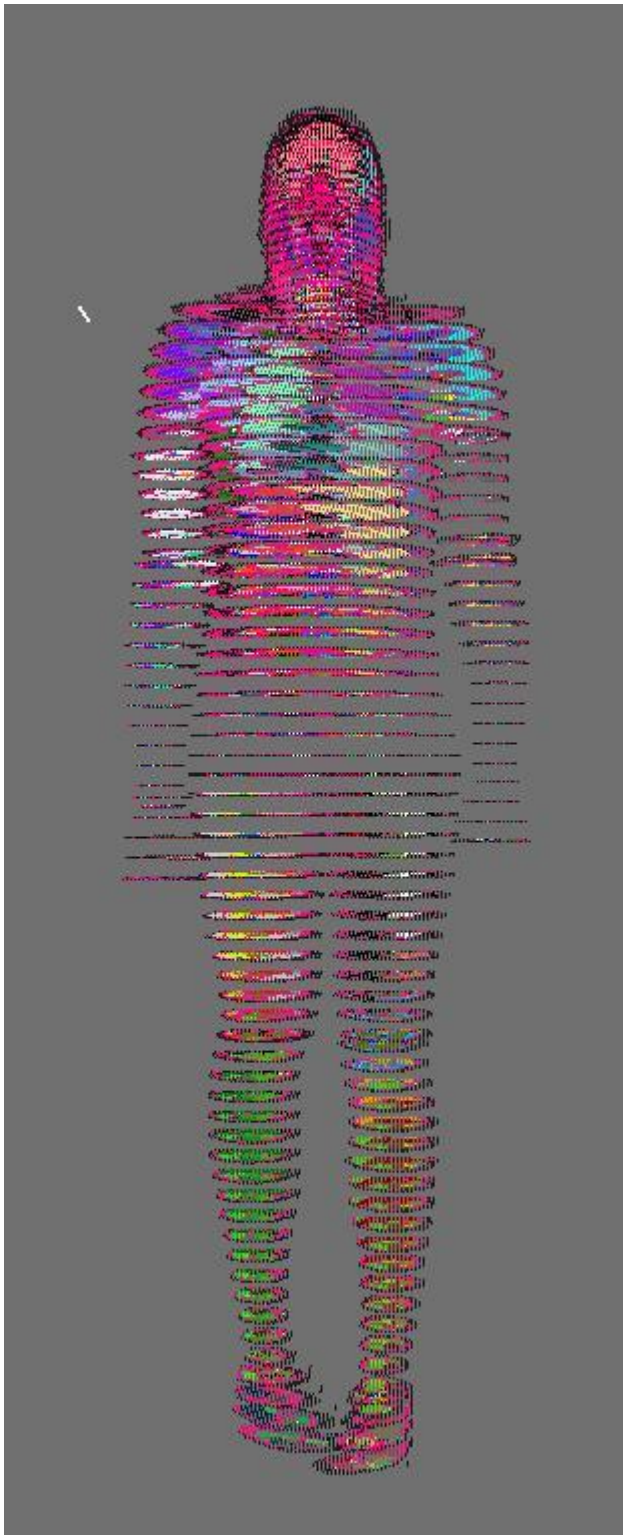

FIGURE 2-3. VH data reshaped as a medium-sized male. The slices are in MRCMan format with points 5mm apart. Slices in the head are 12mm apart, below that they are 26mm apart.

TABLE 2-4. Comparing GEBOD (Col. 1) and ISM (Col. 2) Anthropometry

Dimension	Short Male		Medium Male		Tall Thin Male	
Stature	1510	1512	1778	1746	2032	1954
Head Length	188	175	198	185	201	195
Head Breadth	149	145	151	145	147	145
Shoulder Breadth	365	375	399	432	414	385
Chest Depth	223	216	243	225	199	190
Chest Breadth	295	280	322	305	276	270
Waist Depth	206	215	226	230	170	180
Waist Breadth	276	265	310	290	260	255
Buttock Depth	228	210	248	250	206	215
Hip Breadth	308	280	270	285	277	275
Biceps Circ	316	314	337	331	281	306
Forearm Circ	282	270	304	245	278	291
Thigh Circ	544	549	596	510	489	456
Calf Circ	347	306	378	345	340	336
Crotch Height	692	675	851	884	1024	1014

2.5 SCAN FITTING SOFTWARE DEVELOPMENT

2.5.1 Program objectives. The objective for this task was to adapt the ISM to compute resampled VH data sets that would fit inside the surface contours of 3D human surface scan data sets. Scan data are created by rapid surface digitizers such as are manufactured by Cyberware and TC2. The resulting software had to not only create a resampled data set that matched the dimensions of the scan data, but that could be accurately and at least semi-automatically inserted within the scan 3D surface.

2.5.2 Program description. In order to fit reshaped VH data to a scan, information about the scan dimensions and shape are necessary. The information is then used to reshape the VH data and then to displace the new body segment slice data to the corresponding segments in the scan data set.

For this task, we used a scan of a young man taken at Natick using a Cyberware whole body scanner. The scan data was reformatted as a point cloud using the Natick program NatickMsr, with an i/o function modified by us. The point cloud data was then read into the Beecher program ShapeAnalysis. ShapeAnalysis was used first to pick off surface landmarks which correspond to the body segment axis systems and the segment boundaries used in the ISM. In this way, scan points homologous to the VH data segment slices could be used to find the dimensions to which the VH data must be reshaped.

TABLE 2-5. Surface Scan Landmarks (Ordered)

1. Right Deltoideus	2. Right Lateral Humeral Epicondyle
3. Right Medial Humeral Epicondyle	4. Left Deltoideus
5. Left Lateral Humeral Epicondyle	6. Left Medial Humeral Eipcondyle
7. Right Radial Styloid Process	8. Right Ulnar Styloid Process
9. Left Radial Styloid Process	10. Left Ulnar Styloid Process
11. Cervicale	12. Crotch Point
13. Right Trochanterion	14. Right Lateral Femoral Epicondyle
15. Right Medial Femoral Epicondyle	16. Left Trochanterion
17. Left Lateral Femoral Epicondyle	18. Left Medial Femoral Epicondyle
19. Right Lateral Malleolus	20. Right Medial Malleolus
21. Left Lateral Malleolus	22. Left Medial Malleolus

The program developed for this task is a modification of ism.c called scan_fit.c. The program inputs the surface scan data and landmarks, as well as the VH data and its segment local axis files. Because of their much more varied topography, it was decided not to use the head, hands, and feet. After the user specifies the resolution parameters (slice point distance and slice spacing) for the output files, the program operates in a manner similar to the ISM, computing remodeled VH data to the conform to the scan dimensions, written out in MRCMan format.

1. Input the scan landmark file. This file of 3D points has the orientation of the original scan data.
2. For each segment, first read in the scan data set in its original orientation.
3. Displace the scan data to an orientation defined by the landmarks for the segment being processed. This is comparable to the axis orientation for the comparable VH segment.
4. Measure the scan segment length using the landmarks as defining end points.
5. Beginning at the top of the segment, find the width and depth of the scan segment at increments equal to the spacing between slices selected by the user. Thus, if the user wants the output files to have slices spaced 25mm apart, the function will first find the scan segment width/depth at the top, then measure width/depth down the length of the segment at 25mm decrements.
6. Use the scan segment length and widths/depths to compute scaling parameters used in rescaling each VH slice as in the ISM.
7. Rescale and sample the VH data slices as in the ISM. This will create a file of slice data in the approximate size and shape of the surface scan segment.
8. When the resampled segment is complete, find the landmarks on it that are comparable to those on the scan segment. These will be used to orient this segment data to that of the original surface scan.
9. Repeat for each body segment.
10. Displace the slice data in each body segment file to the scan segment orientation.
11. For each displaced slice in each segment, find the center of the scan data at that point, and translate the slice to that center using the slice centroid.

2.5.3 Test results. Using the one scan data set, the results were fair (Figure 2-4). There needs to be more work to reshape each slice to the exact contours of the surface scan.

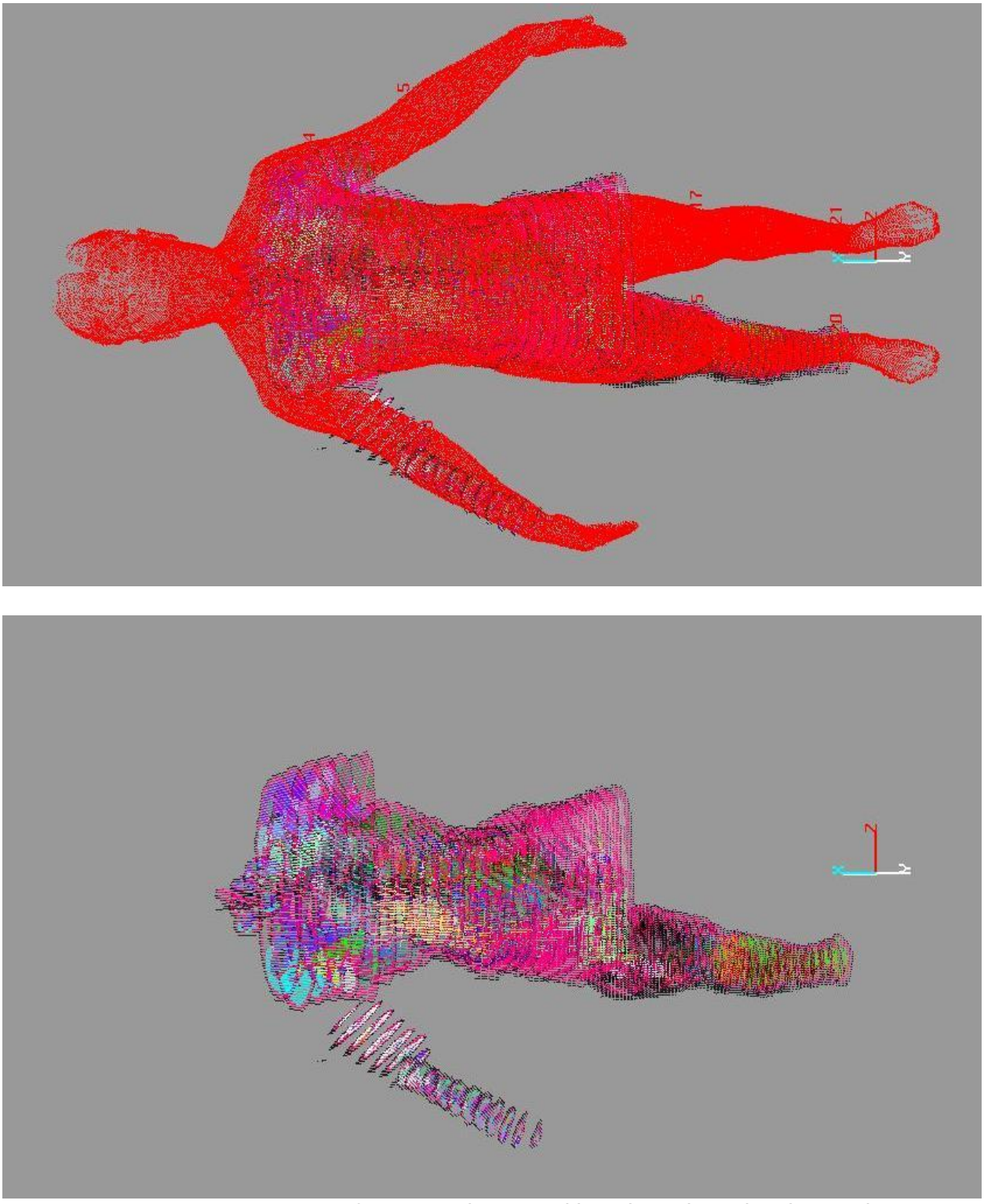

FIGURE 2-4. Top: A surface scan data set with reshaped VH data inserted. Bottom: The VH data without the scan. Due to software limitations, only part of the data can be displayed

SECTION 2 REFERENCES

Clauser, Charles E., Pearl Tucker, John T. McConville, Lloyd L. Laubach, and Joan Reardon, 1972, Anthropometry of Air Force Women, AMRL-TR-70-5 (AD 743 113), Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Gordon, Claire C., Bruce B. Bradtmiller, Thomas D. Churchill, Charles E. Clauser, John T. McConville, Ilse Tebbets, and Robert Walker. 1988 Anthropometric Survey of US Army Personnel: Methods and Summary Statistics, Natick/TR-89/044, U.S. Army Natick RD&E Center, Natick, Massachusetts.

Grunhofer, H.J., and G. Kroh, 1975, A Review of Anthropometric Data on German Air Force and United States Air Force Flying Personnel 1967-1968, AGARD-AG-205 (AD A010 674), Advisory Group for Aerospace Research and Development, 7 Rue Ancelle 92200, Neuilly Sur Seine, France.

McConville, J.T., T.D. Churchill, I. Kaleps, C.E. Clauser and J. Cuzzi, 1980, Anthropometric Relationships of Body and Body Segment Moments of Inertia, AMRL-TR-80-119 (AD A097 238), Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Young, Joseph W., Richard F. Chandler, Clyde C. Snow, Kathleen M. Robinette, Gregory F. Zehner, and Maureen S. Lofberg, 1983, Anthropometric and Mass Distribution Characteristics of the Adult Female, FAA-AM-83-16, Civil Aeromedical Institute, Federal Aviation Administration, Oklahoma City.

3. WOUND BALLISTICS MODELING

3.1 CASUALTY ASSESSMENT

Casualty assessment employed in MRCMan is based on anatomical criteria and is related to the notion of an *operational casualty* and may not relate in an obvious way to the concept of a *medical casualty*. The notion of an *operational casualty* relates to incapacitation that may impair task performance associated with successful completion of a mission. A medical casualty refers to a wound that requires medical attention and may/or may not relate to impaired task performance required for mission completion. The assessment is therefore trimodal. i.e., casualties are categorized as an *operational casualty*, *superficial wound*, or *no wound*.

MRCMan employs the following anatomical criteria for an operational casualty assessment: (1) wounds that penetrate a *critical depth* through soft tissue, (2) projectiles that penetrate into a body cavity such as the cranium, abdomen, or thorax, (3) fracture of long bones, and (4) severing critical blood vessels or nerve plexus. Relative to item 1, each surface voxel of MRCMan has a critical penetration depth associated. If this penetration depth is exceeded then an operational casualty is assessed as occurring. This critical depth of penetration was established by examining the contents of the surface voxels to determine when an important tissue is exposed and its depth. If a wound occurs but penetration is less than the critical depth, then a *superficial wound* is assessed as occurring.

3.2 PENETRATING WOUNDS

3.2.1 Enhanced Projectile Retardation Algorithm

Retardation Force during Penetration of a Projectile in a Material: Inversion of Experimental Penetration Depth Data

During penetration of a projectile inside a target, retardation force that decelerates the projectile arises due to its contact with the surrounding target material. The nature of these contact forces depend on the kinematic state of the projectile, and possibly on its kinematic history as well. The second possibility arises from the influence of the projectile motion on the target at the current location of the projectile due to earlier radiated waves. In the case of a penetration velocity much slower than the speed of sound v_p in the material, retardation force is approximately equal to the compressive fracture threshold of the material, and hence can be regarded as a material constant c_f. On the other hand, when the penetration velocity is comparable to v_p, most target materials behave like viscous fluid media, and hence the force of resistance *F* is no longer a material constant but depends on the local kinematics and possibly on kinematic history of the projectile penetration. Most experimental studies done in the fluidized state of the material assumes that the

character of F depends on the steady state fluid drag on the projectile even when the problem is temporally unsteady. According to this theory, F is proportional to v^2 in the regime where v is the steady state velocity of the flow. But since F is the surface integral of the contact forces over the contact area, it is physically evident that for unsteady problems, such contact area may not only depend on the instantaneous velocity of the projectile but may also depend on the history of loading prior to the current state. Contact area also depends on the locations of boundary layer separations, and the determination of boundary layer separations in unsteady flow is a very complex mathematical problem that has not been solved to date except by purely numerical algorithms and expensive computer programs.

Here we have taken a different approach where our goals are to bypass the complex step of numerical integration of nonlinear differential equations, and to develop a technique based on semi-analytical approach that can be used to estimate the retardation force F for a given target over a given range of entry velocity. For our analysis, we assume that there exists an experimental database that contains an array of penetration depths and corresponding entry velocities for the projectile-target pair under study. Our analysis also deals with the relationships between the model parameters and physical parameters of the problem as much as practicable, so that an extension of our analysis to unknown materials (where no penetration data are available) can be made. With these goals in mind, we start with the assumption that, even in an unsteady problem, F depends only on the instantaneous velocity of the projectile. We also conceive that our *projectile is made up of a rigid collection of particles* so that F can be determined by integrating the forces on such particles over the surface of the projectile. Thus we need a basis of determining the force on one such spherical particle. For our purpose, we use small 'BB's to simulate such particles. Question may arise as to whether 'BB's are suitable for simulating particles that constitute a particular projectile. The success or failure of our approach will determine whether the use of other particle simulants is warranted.

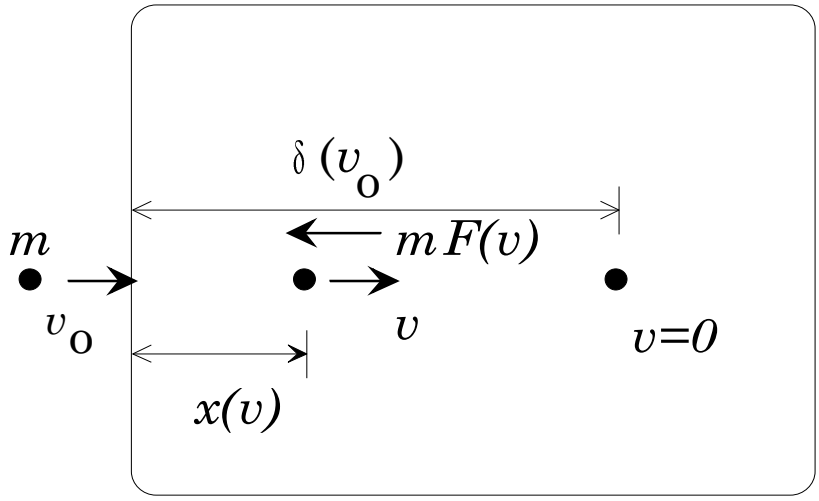

FIGURE 3-1. Schematic of Spherical Projectile (“BB”) Penetrating a Target

Thus we have reduced the problem of determining the retardation force on a projectile penetrating a given target to the problem of determining the retardation force $F(v)$ *per unit mass* on a 'BB' where F is an unknown function of the instantaneous velocity v of the 'BB' inside a target. The problem of penetration of 'BB' inside a target is schematically shown in Figure 3-1.

We also assume that we have an experimental database relating the distribution of the penetration depth function $\delta\ v_0$ (Figure 3-1) as a function of entry velocity v_0. Furthermore, the low and high velocity behavior are also assumed to be known, and these are given by

$$\lim_{v\to 0} F(v) = c_f \ , \quad \lim_{v\to v_p} F(v) = \alpha v^2 \tag{1}$$

In (1), both c_f and α depend on the material properties of the target.

Let us first establish an exact relation between the two function $\delta(v)$ and $F(v)$ introduced above. The equation of motion of the center of mass (CM) of the projectile is given by

$$v\frac{dv}{dx} = -F(v) \tag{2}$$

Integrating (2) from $x=0$ to $x=\delta$, we have

$$\delta(v_0) = \int_0^{v_0} \frac{v dv}{F(v)} \tag{3}$$

Since (3) is valid for all v_0, we have, after differentiation of (3) with respect to v_0,

$$F(v) = \frac{v}{\frac{d\delta}{dv}} \tag{4}$$

Equation (4) is valid for all values of the local velocity v of the projectile inside the target, and establishes the exact relation between the retardation force $F(v)$ ***per unit mass*** and the penetration depth δ. Equation (4) can be used to estimate $F(v)$ from the experimental database of $\delta(v)$. Below we present a method of estimating $F(v)$ from known $\delta(v)$.

Determination of $F(v)$

a. Low Velocity of Penetration

Form the first equation in (1) and (4), it is clear that

$$\lim_{\upsilon \to 0} \delta(\upsilon) \approx \upsilon^2 \tag{5}$$

and hence in the low velocity range $\delta(\upsilon)$ can be expanded in a Taylor's expansion as

$$\delta(\upsilon) = \upsilon^2(a + b\upsilon + c\upsilon^2 + ...) \tag{6}$$

We can now estimate the constants *a, b, c* etc. in (6) by first dividing the experimental data by υ^2 and then fitting a polynomial to the reduced data. The relation between the constant *a* in (6) and the physical constant c_f can be found by using (1), (4) and (6). These give

$$c_f = \lim_{\upsilon \to 0} F(\upsilon) = \lim_{\upsilon \to 0} \frac{\upsilon}{\frac{d\delta}{d\upsilon}} = \frac{1}{2a} \tag{7}$$

Thus the leading term in the expansion of $\delta(\upsilon)$ in (6) can be found from physical data of c_f.

b. High Velocity of Penetration:

From the second equation in (1) and (4), it is clear that $\delta(\upsilon)$ behaves like the natural logarithm of υ when the velocity υ is comparable to the sonic velocity c_p of the target medium. Hence in this region, $\delta(\upsilon)$ can be expanded as

$$\delta(\upsilon) = a' \ln(\upsilon) + \frac{b'}{\upsilon} + \frac{c'}{\upsilon^2} + ... \tag{3.2-8}$$

The constants *a', b', c'* etc. can now be calculated from the experimental data for large υ using a fit of the form of (8) to $\delta(\upsilon)$. Since α in (1) is related to the drag coefficient for the target medium, α is a physical parameter that can be related to the coefficient of a' of the leading term in (8) by using (1) and (8). This yields $\alpha = \frac{1}{a'}$.

c. Solution in the Intermediate Velocity Region

In the intermediate region of the penetration velocity υ, solution for $F(\upsilon)$ and the analytical fit for $\delta(\upsilon)$ can be found by using matched asymptotic expansions of the these functions in the low and high velocity range. Matched asymptotic technique is based on the simple rule that there are two overlapping regions in the intermediate velocity range where the low velocity [Equation (6)] expansion in one region and high velocity expansion [Equation (8)] in the other region yield the same results as that of the intermediate velocity expansion.

Final Polynomial Fit to F(v)

After the determination of $F(\upsilon)$ is complete using the procedure given above, it is possible to fit a polynomial in υ to fit the resulting $F(\upsilon)$. The virtue of such fit can be realized in the

theoretical prediction of the penetration depths for velocities in the interior of or outside the experimental database. Since the calculation of $\delta(v)$ from a fitted $F(v)$ involves an integration (which is a smoothing process) as in equation (3), estimates of $\delta(v)$ are usually very good. These expansions are however not valid for characterization of $\delta(v)$ using the differential relation in (4) as they may predict the incorrect behavior of these functions at both low and high velocity domains. For example, if the polynomial fit

$$F(v) = \beta_1 + \beta_2 v + \beta_3 v^2 + \beta_4 v^3 \tag{9}$$

is used to predict the behavior of $\delta(v)$ for high velocity range using (4), we get

$$\delta(v) \approx \frac{1}{v} \text{ and } F(v) \approx v^3 \tag{10}$$

Equations in (10) contradict (8) and the second equation in (1) respectively.

Data Processing

Low Velocity Region

As indicated before in Section I, in this range we first compute $\frac{\delta(v)}{v^2}$ from the known experimental data prior to a polynomial fit to obtain $\delta(v)$ in the form

$$\delta(v) = v^2 \sum_{i=0}^{n} a_n v^n \tag{11}$$

where *n* is the degree of the polynomial and *a*'s are unknown constants to be determined from the polynomial fit to the experimental data. For *n=2,* we find

$$a_0 = 5.13 \times 10^{-6},\ a_1 = -2.844 \times 10^{-9},\ a_2 = 5.0 \times 10^{-13}$$

High Velocity Region

For high velocities, $\delta(v)$ behaves like $\ell n(v)$ and hence can be expanded as

$$\delta(v) = a' \ln(v) + \sum_{i=1}^{n} b_i' v^{-i} \tag{12}$$

a' and b_i''s can be calculated from fitting (12) to the experimental database for high velocities. This yields $a' = 1.261,\ b_i' = -7.67 \times 10^3$. Comparisons between the low and high velocity

asymptotes and the experimental data are shown in Figure 3-2. As indicated earlier, there is an intermediate region of velocity where both of these asymptotes yield the same penetration depths. For this experimental database, this region is seen to lie between 2000 ft/sec to 2200 ft/sec (Figure 3-2)

Determination of Retardation Force

Using the low and high velocity asymptotes of $\delta(\upsilon)$ given in equations (11) and (12) respectively, and the relation between *F(v)* and $\delta(\upsilon)$ given in equation (4), we can determine the retardation force *F(v)*. For the current database, the computation of *F(v)* is shown in Figure 3-3 by the solid line curve. We fit polynomials of degrees 2, 3 and 4 to the theoretical *F(v)* distribution and these results are shown in Figure 3-3. The coefficients of these polynomial fits are tabulated in Table-3-1. A comparison of the experimental data with the theoretical prediction of the penetration depth function $\delta(\upsilon)$ is shown in Figure 3-4.

TABLE 3-1. Polynomial Fit to the Retardation Force Function F(v)

Degree of Polynomial Fit to F(v)	Coefficients in ascending order of the variable				
2	2.45234×10^{5}	-570.63	0.043681		
3	0.94554×10^{5}	106.4678	2.28712×10^{-5}	7.04×10^{-5}	
4	0.88145×10^{5}	195.33195.	-0.1597	$1.478\times10^{-4}7$	1.072×10^{-8}

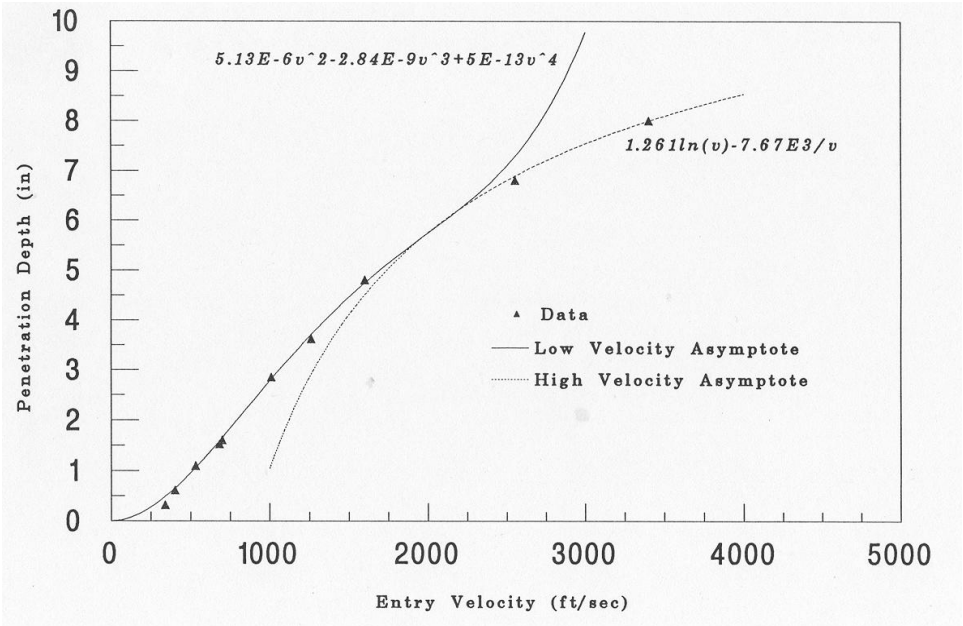

FIGURE 3-2. Comparison between Experimental Data, High and Low Velocity Asymptotes

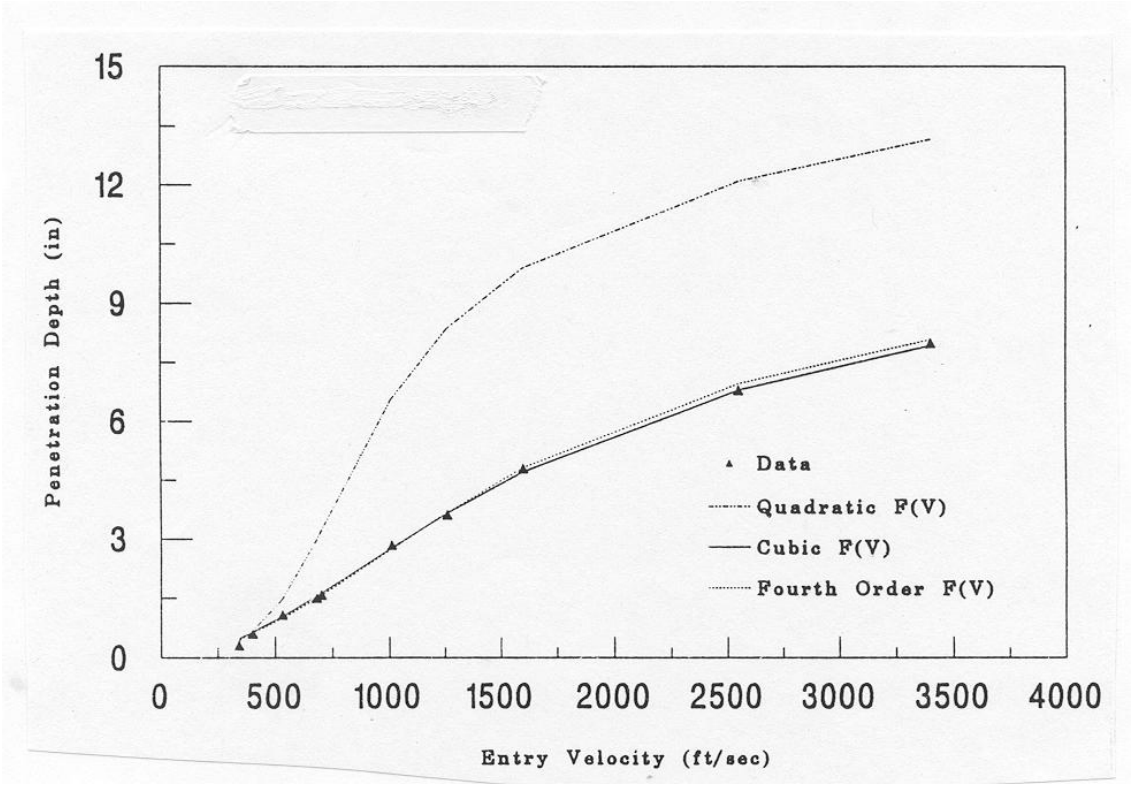

FIGURE 3-3. Theoretical Retardation Force and Various Polynomial Fits

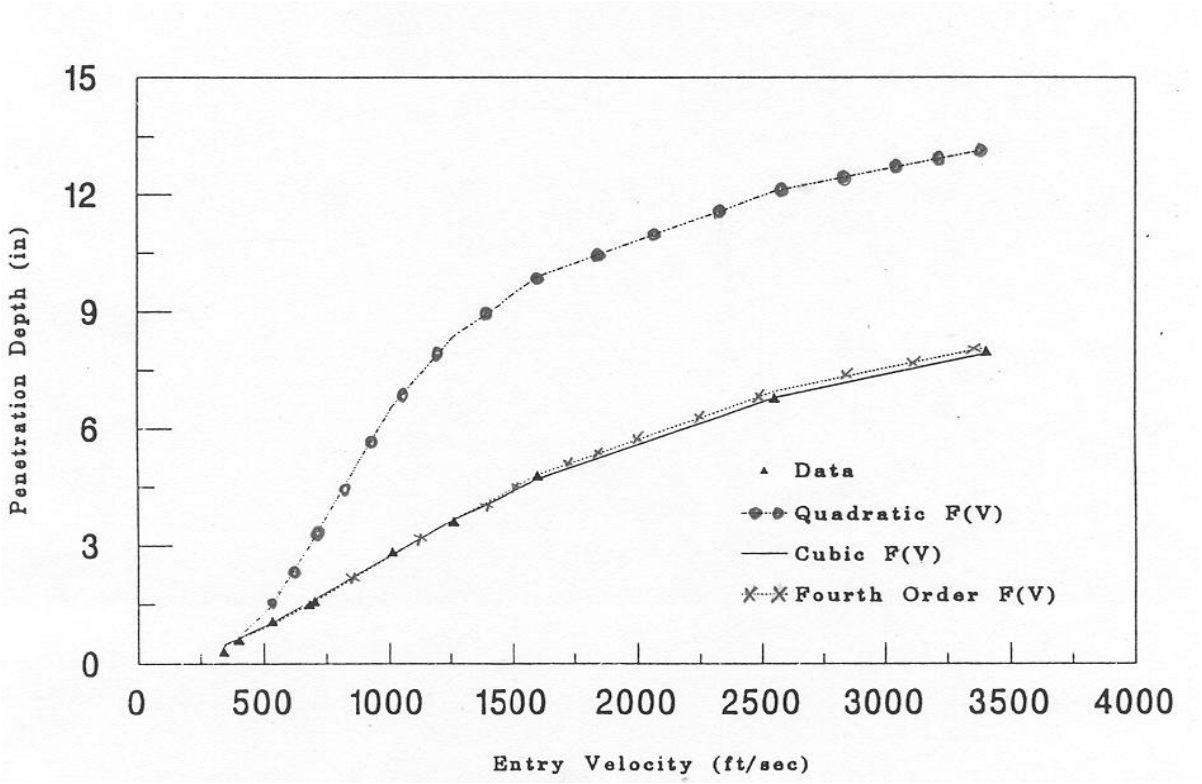

FIGURE 3-4. Comparison Experimental Penetration Data and Theoretical Results

3.2.2 Bone Fracture

Kinematics of projectile penetration in a gelatin with embedded bones incorporate the following bone fracture modes: *drill hole and brittle fracture* which includes spiral and radial fractures, and various types of comminuted fracture. The discrimination of fracture mode is based on an energy criterion – energy per unit area per unit time – and a stress/strain criteria.

Drill Hole Mode

This mode occurs for certain entry velocity of the projectile. The projectile penetrates like a drill hole with clean entry and exit holes.

Brittle Fracture Mode

For certain range of entry velocity, the bone suffers radial fractures that are sometimes so bad that the whole bone may disintegrate. The main mechanism for these type of fractures is the propagation of highly compressive waves through the target. Analytical models have been developed to interpret the existence of and to quantify such radial fractures.

Analysis of Various Modes of Bone Fracture:

It has not yet been conclusively established what parameters of the dynamic process of projectile penetration are responsible for bones to fracture in one mode over others. Two important parameters of penetration dynamics are bullet size and entry velocity. Kinetic energy of the projectile is directly proportional to both of these quantities. From experiments done on gelatin embedded bones where spheres of various sizes are shot, it is seen that increase in sphere size is effective in penetrating the bone with little or no radial fractures on the anterior side while decrease in sphere size is less effective in penetration but effective in creating brittle mode radial fractures. [32] For example, an osteoporotic femur subjected to impact by a 0.406 in. sphere at 600 ft/sec, penetrates the bone with little or no radial fractures while a 0.250 in. sphere at 602 ft/sec did not penetrate but produced radial fractures emanating from the entrance hole. If we increase the speed of the smaller size sphere to 908 ft/sec, the projectile also stops but the bone is broken into two. This case has about 41% more kinetic energy than that of the larger size sphere but no penetration occurs. This shows that the distribution of energy for smaller size spheres is more focused towards radiation of energy than towards penetration. Hence the increase in kinetic energy alone does not necessarily guarantee penetration. We need to know how the projectile size and its velocity affect the distribution of energy.

To calculate the radial fracture from compressive waves, MRC has developed a singular displacement model where, to the lowest order in singularity, the radial displacement decays as $\frac{1}{r}$

[32] Huelke, D. F. et. al, "An experimental study in bio-ballistics: Femoral fractures produced by projectiles-II, Shaft Impacts," *J. Biomechanics*, Vol. 1, pp. 313-321, 1968.

where r is the distance from the center of the projectile footprint. For smaller footprint, it produces more compressive wave for radiation. Some of these singular models are given below. These models give the same qualitative results as observed in the above experiments. For fractures in coated and uncoated polycarbonates, these models yield quantitative estimates of radial fracture that are in extremely good agreement with experiments. However, this mechanism of energy distribution has not yet been completely understood. More systematic parametric studies are needed to resolve this problem.

Radial Cracking Due to Projectile Impact

Radial cracking in a material is caused by the in-plane, tensile stress components normal to the direction of propagating waves parallel to the free surface of the medium. In a cylindrical polar coordinate system (r,θ,z) (Figure 3-5), this stress component is σ_{rr}. In order to derive a simple expression for σ_{rr}, we observe that in the case of a projectile impacting a flat target, the radial displacement field u_r is mostly concentrated near the impact area, hence is possibly mathematically singular at the point of impact, Furthermore, the radial displacement u_r decays with r as we recede away from the point of impact. A simple mathematical expression of u_r having this property can be written as

$$u(r,t) = \frac{a_n}{r^n} + \frac{a_{n-1}}{r^{n-1}} + \frac{a_{n-2}}{r^{n-2}} + ... \tag{13}$$

where *n* is the index of singularity at $r = 0$ and *a's* are functions of time, material properties and impact parameters. Constants *a's* can be calculated from the relevant initial and boundary value problems for this case.

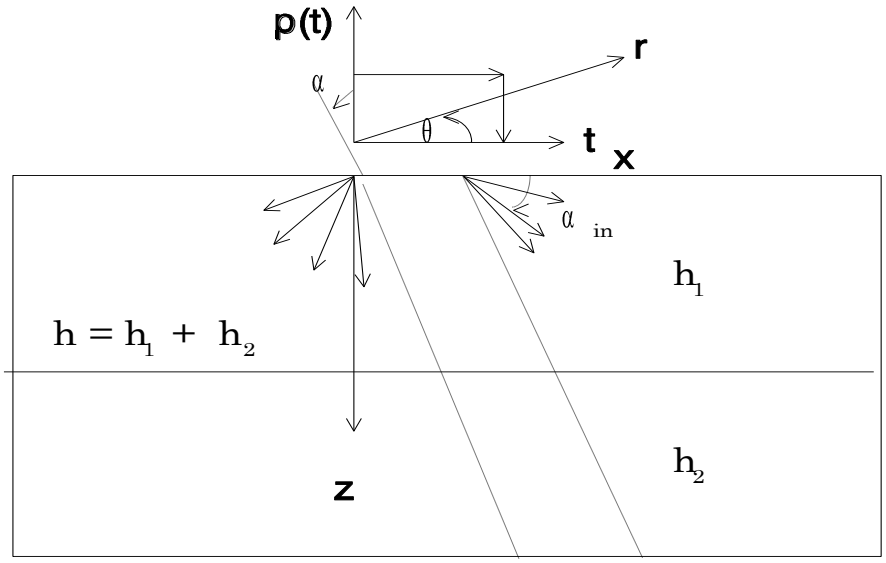

FIGURE 3-5. Emanating Rays from the Impact Points

Some Special Cases

Case 1. As a simple test case, we use the lowest order of singularity by taking n=1 so that

$$u(r,t) = \frac{a_1}{r} \tag{14}$$

Using the stress-strain-displacement relations in the cylindrical polar coordinates we have

$$\begin{aligned} \sigma_{rr} &= E'[(1-\nu)\frac{\partial u_r}{\partial r} + \upsilon \frac{u_r}{r}] \\ \sigma_{\theta\theta} &= E'[(1-\nu)\frac{u_r}{r} + \upsilon \frac{\partial u_r}{\partial r}] \end{aligned} \tag{15}$$

where $E' = \dfrac{E}{(1+\upsilon)(1-2\upsilon)}$, E and υ are the Young's modulus and Poisson's ratio of the material, respectively. Equations 14 and 15 give

$$\begin{aligned} \sigma_{rr} &= -\frac{Ea_1}{(1+\nu)r^2} < 0, \text{ (compression)} \\ \sigma_{\theta\theta} &= \frac{Ea_1}{(1+\nu)r^2} > 0, \text{ (tension)} \end{aligned} \tag{16}$$

From (16), we can conclude that radial cracks are feasible from the lowest order singularity in the displacement field u_r while the circumferential cracks are not feasible. Since the lowest order of singularity is associated with the low velocity impacts, this result is compatible with the experimental observations while radial cracks are always present for such impacts.

Case 2. Now we include the next higher order term in u_r and write

$$u_r = \frac{a_2}{r^2} + \frac{a_1}{r} \tag{17}$$

Equations 15 and 17 yield

$$\begin{aligned} \sigma_{rr} &= E'[(3\nu-2)\frac{a_2}{r^3} + (2\nu-1)\frac{a_1}{r^2}] \\ \sigma_{\theta\theta} &= E'[(1-3\nu)\frac{a_2}{r^3} - (2\nu-1)\frac{a_1}{r^2}] \end{aligned} \tag{18}$$

Since this case corresponds to higher velocity of impact than the previous case, let us explore the possibility of the existence of circumferential cracks while no radial cracks are present. This will represent the reversal of the crack modes observed in case 1 and, to some extent, some experimental data we have for high velocity of impacts. To do so, we need to make

$\sigma_{rr} < 0$ and $\sigma_{\theta\theta} > 0$. From (18), this requires a_1 to be negative and a_2 to be positive. Besides, the Poisson's ration υ should lie between $\frac{1}{3} < \nu < \frac{1}{2}$. For polycarbonates, υ is approximately 0.38 which lies in this range. To put this result in a better perspective, we rewrite the displacement field in this case as

$$\frac{u_r}{a} = a_2\left(\frac{a}{r}\right)^2 - a_1^{'}\left(\frac{a}{r}\right) \tag{19}$$

where a is some characteristic length scale in the problem, $a_1^{'} = -a_1$. For the polycarbonate impact problem under study, we may use the projectile foot print radius as a. In this form, both $a_1^{'}$ and a_2 have the dimension of length and both quantities are positive. To ensure no negative radial displacement, we need a_2 to be much larger than $a_1^{'}$. Such distribution ensures that $\sigma_{\theta\theta} < 0,$ and $\sigma_{rr} > 0$ beyond some r. For a given tensile stress threshold σ_c for the occurrence of circumferential failures, the radius r_c of the zone of no tensile circumferential stress can be calculated from the condition that $\sigma_{rr}(r = r_c) = \sigma_c$ which gives, from (15) and (19),

$$a_2(3\nu - 2)\left(\frac{a}{r}\right)^3 + a_1'(1 - 2\nu)\left(\frac{a}{r}\right)^2 = \frac{\sigma_c}{E'}$$

When $\sigma_c = 0,$ r_c is given by

$$\frac{r_c}{a} = \frac{(2 - 3\nu)}{(1 - 2\nu)}\left(\frac{a_2}{a_1'}\right) \tag{20}$$

For example, when υ = 0.38, r_c = 3.58a which is in a region where u_r < 0. Besides, as observed before, $a_1^{'}$ is small compared to a_2, and hence we see that radial displacement model do not predict the occurrence of circumferential cracks in coated polycarbonates. This is due to the fact that tensile circumferential stress responsible for the circumferential fractures, are generated due to wave transmission and reflection inside the composite, and hence can be explained only by analysis similar to the wave ray analysis discussed in the previous section.

Radial and Circumferential Fracture from Singular, Impact Induced Elastodynamic Displacement Fields: Nondimensional Analysis

Generation of radial and circumferential cracks from singular stress fields depend on the nature of singularity in the displacement fields. Except for the case of shock induced elastic fields, stresses can be calculated from the displacement field from the usual stress-displacement relations. Below we present a description of various possible singular elastic fields that can be generated due to impact of a projectile with a target. Assuming that the elastic displacement can be expressed as

$$u_r = \frac{a}{r^{\alpha}} \tag{21}$$

where u_r is the symmetric radial displacement in a cylindrical polar coordinate system. All other displacement components are assumed to be zero, and α is a real positive number which gives the strength of singularity in the displacement field. Equation 21 yields the normal stress components as

$$\sigma_{rr} = \frac{Ea[(\alpha+1)\upsilon - \alpha]}{(1+\upsilon)(1-2\upsilon)r^{\alpha+1}}$$
$$\sigma_{\theta\theta} = \frac{Ea[1-(\alpha+1)\upsilon]}{(1+\upsilon)(1-2\upsilon)r^{\alpha+1}} \qquad (22)$$

For tensile stress induced failures, we need, for radial cracks, $\sigma_{\theta\theta} > 0$, and for circumferential cracks $\sigma_{rr} > 0$. For both failure modes possible, (22) gives

$$(\alpha+1)\upsilon - \alpha > 0$$
$$1-(\alpha+1)\upsilon > 0 \qquad (23)$$
$$\text{so that } \frac{\alpha}{1+\alpha} < \upsilon < \frac{1}{1+\alpha} \text{ which requires } \alpha < 1$$

For failures in radial crack mode only, we need only the second equation in (23) to be satisfied. Since this requires $\upsilon < \frac{1}{1+\alpha}$, radial cracks are always possible in polycarbonates while circumferential cracks are possible for displacement fields with singularity index of $\alpha = \frac{1}{2}$ (shock-induced field) since υ lies between 1/3 and 1/2 for polycarbonates.

Force of Resistance during the Early Phase of Bone Penetration:

In order to understand the nature of the force of resistance during the early phase of penetrating a material discontinuity, let us consider the following problem (Figure 3-6). Let the projectile was moving with a velocity ***v*** in the *x*-direction just before it contacts the bone. We assume that at the point of contact on the surface of discontinuity between the gelatin and a bone, the common normal makes an angle θ with the direction of the pre-entry velocity of the bullet. We also assume that the force of resistance *F* acts in the direction of common normal (this is a condition that we can later relax). Since the velocity of the bullet suffers discontinuity, let φ be the angle made by the post-entry velocity ***v'*** of the bullet. From Newton's laws of motion applied on the center of mass of the bullet, we have

$$m(\vec{v}' - \vec{v}) = \vec{F}\Delta t \qquad (24a)$$

Equation 24a gives

$$v_x' = v - \frac{F}{m}\cos\theta\Delta t$$

$$v_y' = \frac{F}{m}\sin\theta\Delta t \qquad (24b)$$

$$\tan\phi = \frac{\frac{F}{m}\sin\theta\Delta t}{v - \frac{F}{m}\cos\theta\Delta t}$$

In the above equations, Δt is the transition time from $\vec{v}$ to $\vec{v}'$.

If F is of the order O(v^2), then $\phi = \pi - \theta$ and we have a complete rebound of the bullet. Thus for penetration *F is at least O(v)* which gives the new angle φ as

$$\phi = \frac{\frac{\alpha}{m}\sin\theta\Delta t}{1 - \frac{\alpha}{m}\cos\theta\Delta t} \qquad (24c)$$

$$F = \alpha v$$

If *F = o(v)*, then φ = 0 so that bullet moves with unchanged direction.

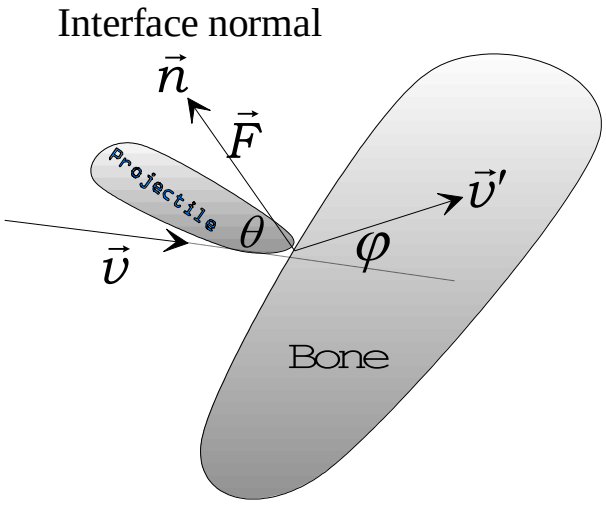

FIGURE 3-6. Kinematics of Pre- and Post-Entry Velocities

Using the above methods, MRC has developed analytical models capable of calculation wound track geometry for motion of a bullet or other similar bodies of high aspect ratio in a bone-tissue environment. It is assumed that the bullet suffers no rotation until it hits the bone so that its center of mass moves in a straight line. It is possible to relax this condition on the projectile

kinetics but the resulting problem becomes three-dimensional. Besides, for most problems of interest, this condition does not impose any limitations in applications of these models. In our model, the target is composed of tissue matrix with an embedded bone. Here the tissue material is replaced with a 20% gelatin or any other suitable medium where the characteristic of the force of resistance as a function of instantaneous velocity is known. The method of determination of the force of resistance has been discussed before. We should recall at this point that the coefficients appearing the description of this force of retardation are influenced by the material properties of target and projectile as well as by the geometrical shape of the bullet tip. In our earlier analysis, we have developed a sufficiently accurate model that describes the force of retardation as cubic or fourth order polynomial in the instantaneous velocity v. For projectiles impacting bones, depending on the entry velocity v_c of the projectile prior to hitting bone the bone, various modes of motion can be classified as follows. (1) *No penetration of bone.* In this case, v_c is not enough to penetrate so that the bullet either turns or rebounds. This threshold velocity v_d is approximately 200 ft/sec for osteoporotic bones and 700 ft/sec for normal bones; (2) *Drill hole mode:* This mode occurs for certain entry velocity of the bullet. Bullet penetrates like a drill hole with clean entry and exit holes; and (3) *Brittle fracture mode:* For a given bullet and certain range of entry velocity , the bone suffers radial fractures that may completely disintegrate the bone. The main mechanism for these fractures is the propagation of highly compressive and reflected tensile waves through the target.

We have successfully developed models which uses an already existing two-dimensional code for determining the wound track in homogeneous medium. This code is first used to determine the motion of the bullet just prior to hitting the bone. If the bullet velocity at this instant is below v_d, the bullet will either rebound or turn. This produces some discontinuities in the bullet velocity (both translational and angular). These discontinuities have been calculated using impulse equations and then the data on linear and angular velocity of the bullet is fed to the above two-dimensional code as initial values for the calculation of the subsequent wound track geometry.

The case of the drill-hole mode can also be handled the same way using the same code as above. In this case, however, the bullet continues to move in a straight-line path after entering the bone with different retardation coefficients than that of a gelatin target.

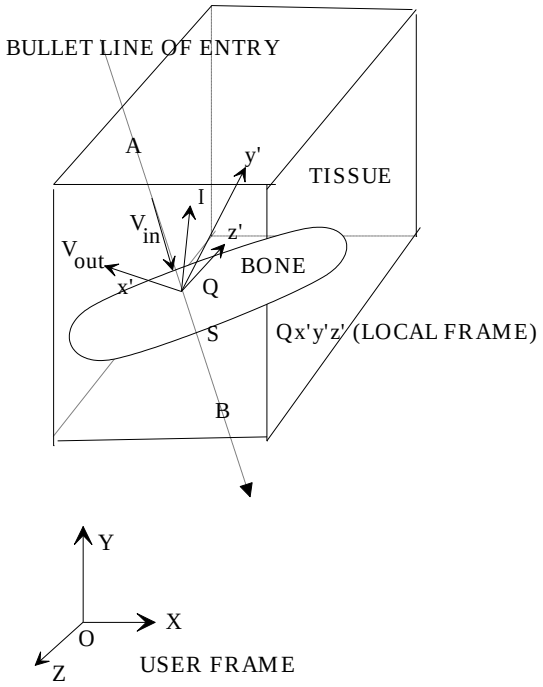

FIGURE 3-7. Schematics of BONEGEL Code

A possibility of a critical shear force model has also been tested for applications in the cases of drill-hole mode fractures for low velocities. Experimental data shows that for the osteoporotic bones, the average force of resistance is 11.68 lb. for 0.250-in diameter spherical projectiles while it is 12 lb. for 0.406-in diameter projectiles moving at 200-400 ft/sec. This data justifies a critical force criterion for low velocity penetration of projectiles in bones.

Schematics of the model used to develop BONGEL Code

A schematic of the model used in the development of BONGEL code is shown in Figure 3-7.

Legend for Figure 3-7.

A(XA,YA,ZA) – Bullet Body Entry Location
Q (XQ,YQ,ZQ) – Bullet Bone Entry Location
S (XS,YS,ZS) – Bullet Bone Exit Location
B (XB,YB,ZB) – Bullet Body Exit Location
ℓ_1, m_1, n_1 = (L1,M1,N1) = Direction cosines of bullet entry line AB with respect to the user coordinate system OXYZ.
ℓ', m', n' = (LP,MP,NP) = Direction cosines of the rebound velocity $\underset{\sim}{V}_{out}$

Notes
All coordinates are in the user-defined coordinate frame OXYZ
For the local frame $Qx'y'z'$, Q'_x, Q'_y lie on the same plane that contains the bone incident velocity $\underset{\sim}{V}_{in}$ and the impulse vector $\underset{\sim}{I}$
Various Phases of Motion: V1CR, V2CR are the two critical velocities used in the code. They control the codes as follows:

A. Rebounding occurs after bone impact when $V_{in} <$ V1CR. This mode gives nonlinear path of the bullet. The bullet spins and moves as it penetrates the target.
B. Clean-hole-bullet-penetration occurs when $\text{V1CR} < V_{in} < \text{V2CR}$. No bone fragmentation occurs in this mode. Linear path, no bullet spinning occurs.
C. Bone fragmentation occurs when $V_{in} >$ V2CR. Linear path, no bullet spinning occurs.

Description of the two input angles THD(θ_d) and PHD(φ_d) used in the BONGEL code are shown in Figure 3-8.

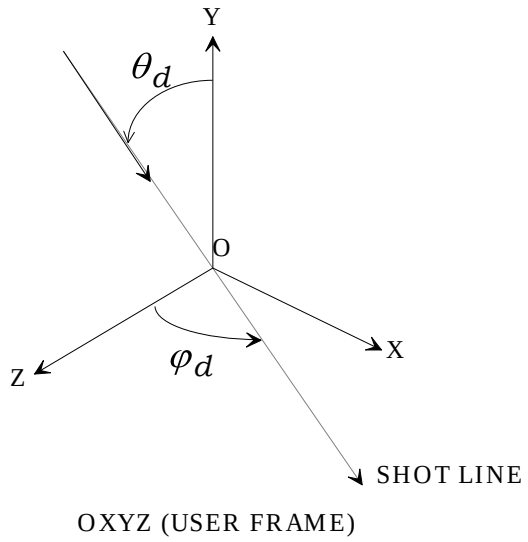

FIGURE 3-8. Description of Projectile Shotline

3.2.3 Curvilinear Projectile Trajectory

Mathematical Model of a Two-dimensional Motion of a Flechette inside a Viscoelastic Fluid

Bodies penetrating a highly viscous material suffer retardation forces mainly originating from three fundamental sources. These are (a) dynamic pressure drag force which is usually proportional to the square of the instantaneous velocity v of the contact point between the projectile and the target, (b) viscous drag force proportional to the magnitude of v and (c) a drag independent of v. The last term is the force needed to fracture the material in the quasistatic mode of target penetration. From a dimensional point of view, independent dimensions of both v^2 and v can be found from the physical properties of the target and the projectile, and hence any power of such non-dimensional groups can be deemed to influence the retardation terms. Obviously, the simplest function of these non-dimensional groups is a polynomial function of the form av^2+bv+c where a, b, c are constants depending of the size, shape and other material properties of both the projectile and the target. Determination of these constants for a specific case can be done from low-velocity and high-velocity experimental data, and by using a matched asymptotic expansion for the intermediate velocity regime.

For 'BB' like spherical projectiles, experimental data have been analyzed and discussed earlier. Since the motion of the center of mass depends only on the vector sum of all the forces acting on a rigid body, we can use this information to estimate the force on a point mass moving in the same media when the instantaneous velocity of the point mass is known. Using this concept, a mathematical model has been developed which is capable of predicting the nature of the two-dimensional motion of a flechette inside the target when the initial conditions are prescribed. Projectiles are not allowed to deform as they penetrate a target, and hence can be regarded as a

rigid body. The motion of the flechette is divided into two parts; motion of the center of mass and the motion about the center of mass. Since the order of these two modes of motion is immaterial for a rigid body, the location of a flechette inside the target at any time can be uniquely computed from analyses of these two modes of motion.

In the current MRCMAN code, the motion of the Center Of Mass(CM) of a projectile entering a target is calculated by integrating the differential equation of motion in the velocity-distance space under the assumption that the retardation coefficients *a, b, c* are known. It should be pointed out that these coefficients represent some average values of the retardation coefficients, and can be used to determine the motion of CM only. To extend the results to full two-dimensional prediction of flechette motion inside a target, we need to make some postulations. One of the most important postulation that we have used here in our model, is that these coefficients are also capable of estimating the force of retardation on a point mass moving inside a target. If this is true, then a rigid body may be conceived of as an ensemble of particles on each of which the retardation force is known, and hence the equations of motions governing the motion of and about the CM can be written down. If we ignore the frictional drag on these particles, then the force of resistance can be calculated by the same functional form shown above except that v should be replaced by the normal component of the velocity of a particle mass on the projectile. Here the normal direction is defined by the outward normal direction at the projectile surface.

Assuming that the ***force of retardation R(v) per unit mass*** on each of these collection of particles constituting the projectile, we can write *R(v)* as

$$R(v) = av^2 + bv + c \tag{25}$$

where the v is the normal component of the velocity as described above. The distribution of these forces on the body and fin area of a flechette is shown in Figure 3-9.

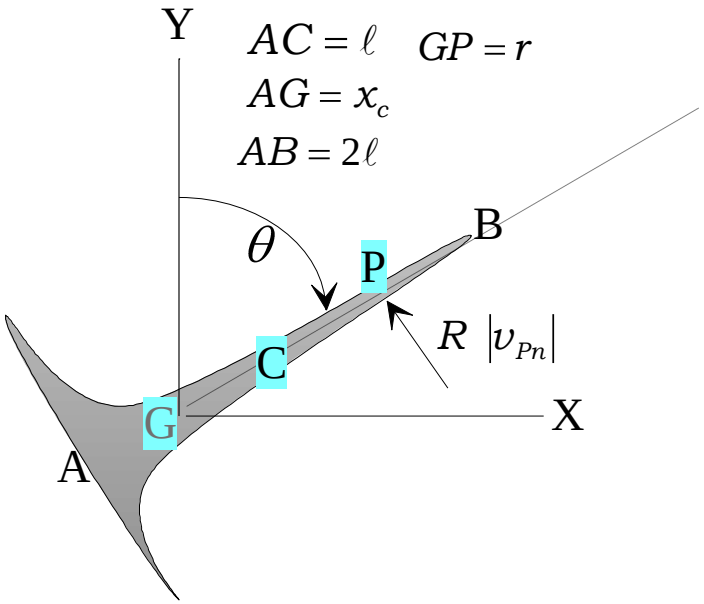

FIGURE 3-9. Schematics of Flechette Kinematics

Due to this shift in the center of mass, the equations of motion of the flechette for the motion of and about the center of mass are modified as follows:

$$m\ddot{x}_G = -\int_{-x_c}^{2l-x_c} \rho R(|v_{pn}|)\,\mathrm{sgn}(v_{pn})\cos\theta\, dr - m_f R(|v_{en}|)\,\mathrm{sgn}(v_{en})\sin\theta$$
$$m\ddot{y}_G = +\int_{-x_c}^{2l-x_c} \rho R(|v_{pn}|)\,\mathrm{sgn}(v_{pn})\sin\theta\, dr - m_f R(|v_{en}|)\,\mathrm{sgn}(v_{en})\cos\theta \qquad (26)$$

and

$$I_G\ddot{\theta} = -\int_{-x_c}^{2l-x_c} \rho R(|v_{pn}|\,\mathrm{sgn}(v_{pn})\, r\, dr \qquad (27)$$

In (25) and (26), the distance x_c between the fin-end of the flechette and the center of mass *G* is given by

$$x_c = \frac{l}{1+\frac{m_f}{m_b}} \qquad (28)$$

The moment of inertia I_G about the shifted center of mass *G* is given by

$$I_G = m_f x_c^2 + m_b[\frac{l^2}{3} + (l - x_c)^2] \tag{29}$$

For fin mass m_f small compared to the body mass m_b, $x_c \approx l$ which reduces equations (21) through (27) to the equations of motion used previously. For the flechette used in our analysis, the body mass m_b (*3.6 gm*) is about seven times larger than the fin mass m_f (*0.5 gm*). Thus these corrections are not going to affect the kinematics description of the center of mass significantly unless the angular velocity of the flechette is very high.

Comparison with the Experimental Data:

Some experimental data is available where flechettes are shot into 20% gelatins. Some of these data are not usable because they are either incomplete or obscure or flechettes have left the target medium so that no data on penetration depth is available. In this report we try to correlate some of these experimental data with the theoretical prediction obtained from the above analytical modeling.

Before we present this comparison, let us point some of the aspects of the analytical modeling and how data is extracted from these and other experiments so as to estimate the kinematic and kinetic properties of the flechette and the gelatin medium. It is assumed in the above analytical model that the force of resistance can be well represented by a polynomial in the instantaneous velocity of the point of contact of the target with the medium. For a long, distributed body (as we have here in the case of a flechette), any point have two velocity components. If we assume that the frictional force can be ignored, the force of resistance can be assumed to be normal to the penetrating body and is a function of the instantaneous normal velocity component of the point. This from of penetration is shown in Figure 3-10 below. Besides, we have shown in a separate study (reported earlier) that for a penetrating sphere, the force of resistance can be well represented by a quadratic (or cubic which is better) function of the instantaneous velocity. In deriving the force of resistance associated with each segment of the flechette, we have assumed that a flechette is composed of a continuous distribution of particles over the geometry of the body so that the total resistance to the flechette motion is coming from the distribution of such forces on the flechette. The integrals appearing in the equations of motion of the center of mass and for motion about the center of mass, represent the total contribution of these force of resistance.

For shot number 15, intermediate velocity data is also available and these data are compared with the theoretical results in Figure 3-12. Here the correlation is reasonably good.

Theoretical results on the calculation of yaw angles versus penetration depths are not exactly correct due the incorrect limits of integration used in Equations (26) and (27) above. However, since the body mass of the flechette is large compared to its fin-mass, results are approximately correct. But due to some bug in the current version of the code, yaw results suffer from some instability in the time domain and we are currently trying to fix this problem. For the case where the force of resistance is proportional to the square of the velocity i.e., $a \neq 0, b = c = 0$ which

corresponds to the aerodynamic model used in the 'ComputerMan' Code, the current version of the flechette code is working, and the result for the yaw angles for shot 15 is compared with the data in Figure 3-12. Here again, the correlation between the analytical prediction and experimental result is reasonably good.

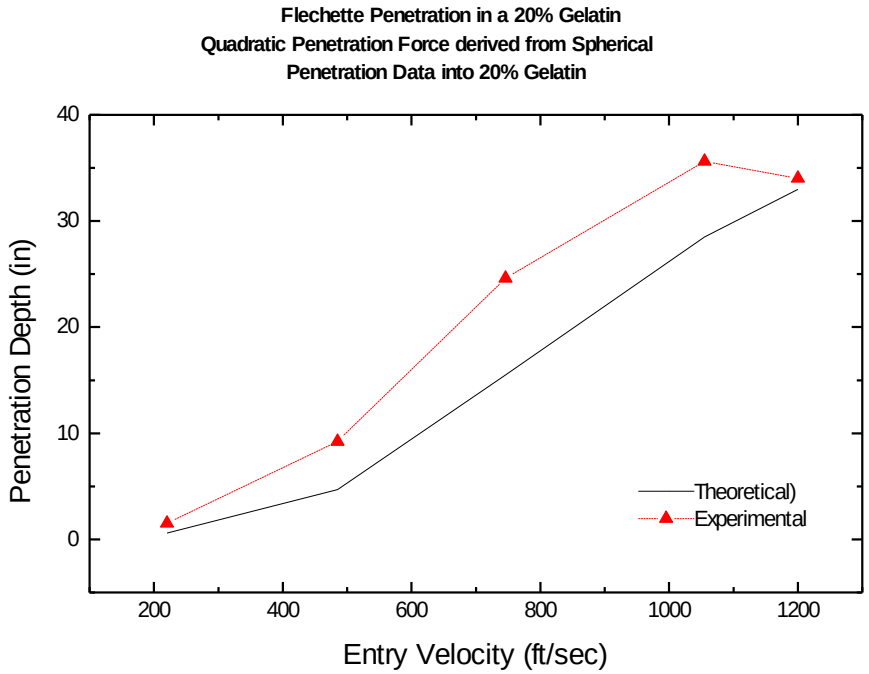

FIGURE 3-10. Theoretical and Experimental Penetration Depths

For better execution of the flechette code, the modified version of the flechette kinetics discussed here should be incorporated in the code. At this point, we estimate that this requires about one man-month work. We have already developed a better numerical algorithm for implementing the modified results, where the integrals will be evaluated directly by using Gaussian, quadrature algorithm so that we may bypass the need for tracking the sign reversals in the integrals used in the current version of the flechette code.

Extension of CM Database to Other Projectiles and Materials

Assuming that the dependency of the model constants a_{val}, b_{val} and c_{val} on the projectile shapes and material, and the target, are expressible in variable separable forms shown in (3.2.2-18), it is possible to extend the current database to other projectiles e.g., flechettes penetrating a human body. Let us first present the mathematical background of such extension. Later we will present some examples of these extensions and compare them with available results.

Let $c_{k,p,m}$ represents a_{val}, b_{val} and c_{val} for $k = 1,2,3$ respectively and subscripts p and m indicate the relevant projectile and target properties. Then according to the variable separable form, we have

$$c_{k,p,m} = f_{1k}(p) f_{2k}(m) \tag{30}$$

In (30), $f_{1k}(p)$ and $f_{2k}(p)$ are functions of the relevant projectile and material properties respectively. At this point, it is not essential to be specific about what properties are involved in shaping these functions. In the following, we assume that the c's in (30) are known either from the current CM database or from other experiments.

Let us consider two separate experimental data involving one projectile type *p* and two target types m_1 and m_2. Then

$$\begin{aligned} c_{k,p,m_1} &= f_{1k}(p) f_{2k}(m_1) \\ c_{k,p,m_2} &= f_{1k}(p) f_{2k}(m_2) \end{aligned} \tag{31a}$$

so that

$$\frac{f_{2k}(m_1)}{f_{2k}(m_2)} = \frac{c_{k,p,m_1}}{c_{k,p,m_2}} \tag{31b}$$

The ratio in (31b) is independent of the projectile type p.

Similarly, if we do two experiments with the same target but two different projectiles, we have

$$\frac{f_{2k}(p_1)}{f_{2k}(p_2)} = \frac{c_{k,p_1,m}}{c_{k,p_2,m}} \tag{31c}$$

The ratio in (31c) is independent of the target material,

Let us now see how the above equations can help us in extending the database to other projectiles and materials. We consider a specific example but the result is general for other usage. It has been pointed out before that the current CM database includes four different projectile shapes (spheres, cubes, cylinders and fragments) while the target materials are divided into 9 types shown in Table 3-2. Let us consider the problem of extending these results to the case of a

flechette penetrating any of the nine types of body parts given in Table 3-1. To be specific, let us extend the database to the case of a flechette penetrating a bone, and this is indicated by a superscript 4 in the equations below. The superscript 1, 2 and 3 refers to the case of projectile-target pair (sphere,bone), (sphere,gelatin) and (flechette,gelatin) , respectively.

Using (30) for the retardation constant a_{val} for the above three pairs for which data is available, we have

$$a^1_{val} = f_{11}(sphere) f_{21}(bone) \tag{32a}$$

$$a^2_{val} = f_{11}(sphere) f_{21}(gelatin) \tag{32b}$$

$$a^3_{val} = f_{11}(flechette) f_{21}(gelatin) \tag{32c}$$

Equations (32) give

$$a^4_{val} = f_{11}(flechette) f_{21}(bone) = \frac{f_{21}(bone)}{f_{21}(gelatin)} \bullet f_{11}(flechette) f_{21}(gelatin) = \frac{a^1_{val}}{a^2_{val}} a^3_{val} \tag{33}$$

Equation (33) yields the required extension. b_{val} and c_{val} can also be calculated similarly.

Modifications, Refinements and Extensions of Current ComputerMan Database

MRCMAN uses a modified database that has been derived from original ComputerMan (CM) by using the following algorithm. The retardation model used in the ComputerMan (CM) code has three constants that depend on the type of projectile and target. It determines the motion of the center of mass of the projectile inside the target using the equation of motion of the form

$$m_p v \frac{dv}{dx} = -R(v) \tag{34}$$

where $R(v)$ is the force of resistance that is related to the instantaneous velocity v through

$$R(v) = \left(\frac{m_p}{\rho_p} \right)^{1/3} (a_{val} v^2 + b_{val} v + c_{val}) \tag{35}$$

where the subscript p is related to the properties of the projectiles, m is the mass and ρ is the density.

The values of a_{val} and c_{val} used in the CM database and its extension to flechette are given in Table 3-2. For all of these cases, $b_{val} = 0$. For the case of flechette penetration in gelatins, the extension of the database for the values of the retardation coefficients has been made using the Equations 30 and 31.

TABLE 3-2. Retardation constants in Extended CM database

Model Constants ⇨		a_{val}				c_{val}		
Projectile Type ⇨	**Sphere**	**Cube**	**Cylinder**	**Flechette**	**Sphere**	**Cube**	**Cylinder**	**Flechette**
Body Location ⇩								
Skin	0.1226	0.2007	0.2117	0.0221	0	0	0	0
Face, Heart	0.192	0.4536	0.3129	0.0346	0	0	0	0
Pancreas, Kidney	0.2422	0.4593	0.3361	0.0436	0	0	0	0
Lung	0.1577	0.4037	0.3115	0.0284	0	0	0	0
Skull, Vertebra, Bone	0.4545	0.834	0.5158	0.0818	753.6E6	761.2E6	9.21E+08	135.6E6
Sternum, Joint, Femur	0.3035	0.5809	0.3677	0.0546	234.1E6	216.6E6	2.96E+08	23.4E6
Brain, Eye, Spinal Cord	0.2059	0.483	0.35	0.0371	0	0	0	0
Scapula, Tibia	0.7418	1.137	1.111	0.1335	1257E6	801.7E6	1.31E+09	226.3E6
Pharynx, Larynx	0.2247	0.7889	0.4234	0.0404	527.3E6	444.0E6	1220E6	95.0E6

Application to Flechette Penetration Database

We have some experimental results on the penetration of flechettes into 20% gelatins. MRC has developed a flechette code which is capable of determining the two-dimensional motion of a flechette when the quadratic form of the retardation force is prescribed. To estimate the accuracy our flechette code which is based on modeling the projectile as a rigid collection of particles where the resistance to particle penetration is derived from small spherical projectiles, we have applied the above estimated polynomial fit to the retardation force function as input to our flechette code. Unfortunately, at the time of development of the flechette code, we assumed that a good quadratic fit is possible, and hence a quadratic retardation force vs. local velocity is assumed in the code. Since it is somewhat time-consuming at this point to modify our flechette code to incorporate higher order polynomial fits, we decided to use the modified coefficients for the quadratic *F(v) vs. v* (Figure 3-11) in our flechette code to obtain and compare the theoretical predictions to the experimental flechette database. For various experimental shot events, the results are shown in Figures 3-12 through 3-14. The results seem to agree reasonably good considering the fact that we have not used a good fit to the retardation function *F(v).* In future, when the current flechette code will be modified to accept higher order of *F(v) vs. v* polynomials, we expect a much better agreement between the experimental data and theoretical results.

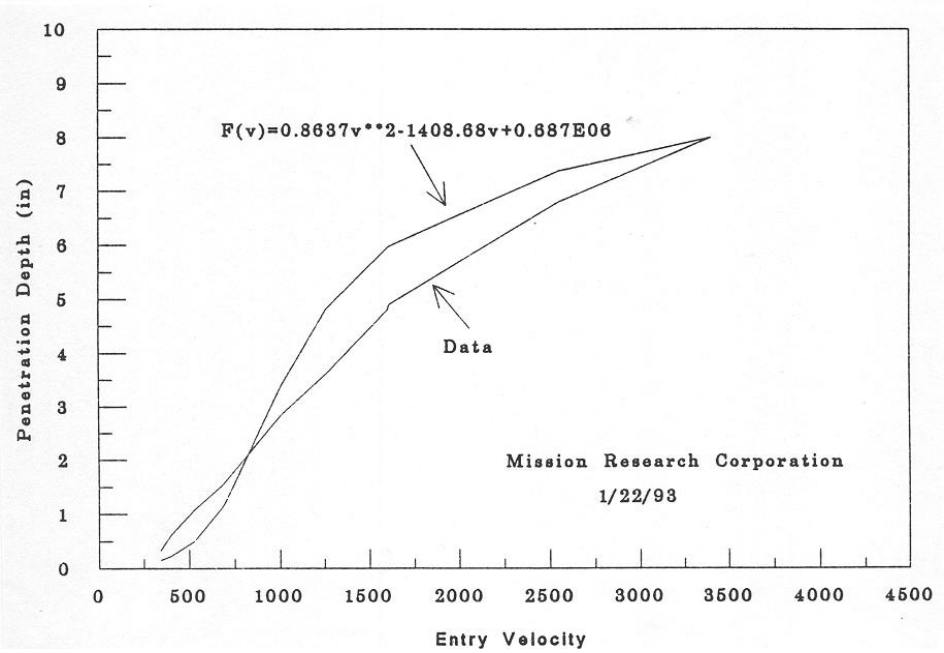

FIGURE 3-11. Experimental data and best theoretical fit with quadratic force function

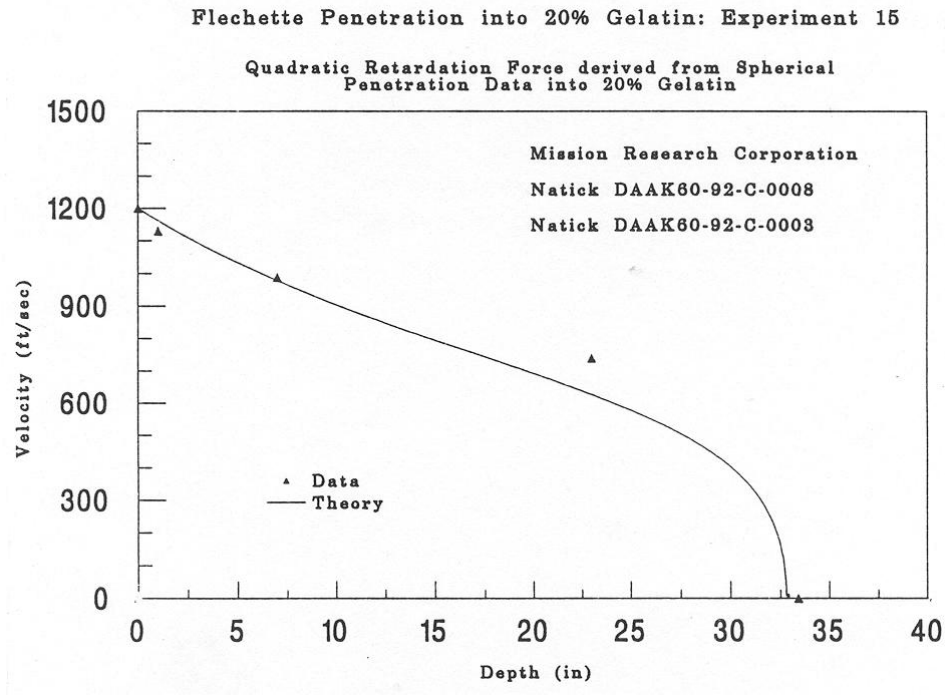

FIGURE 3-12. Flechette penetration into 20% gelatin: Theory and Experiment: Test 15

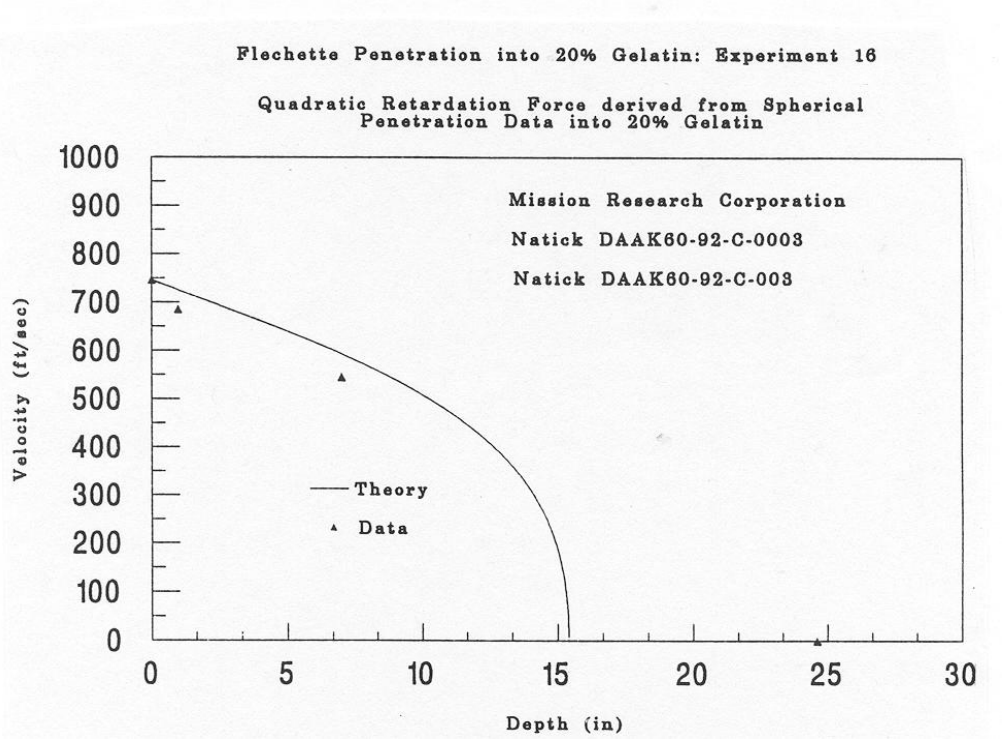

FIGURE 3-13. Flechette penetration into 20% gelatin: Theory and Experiment: Test 16

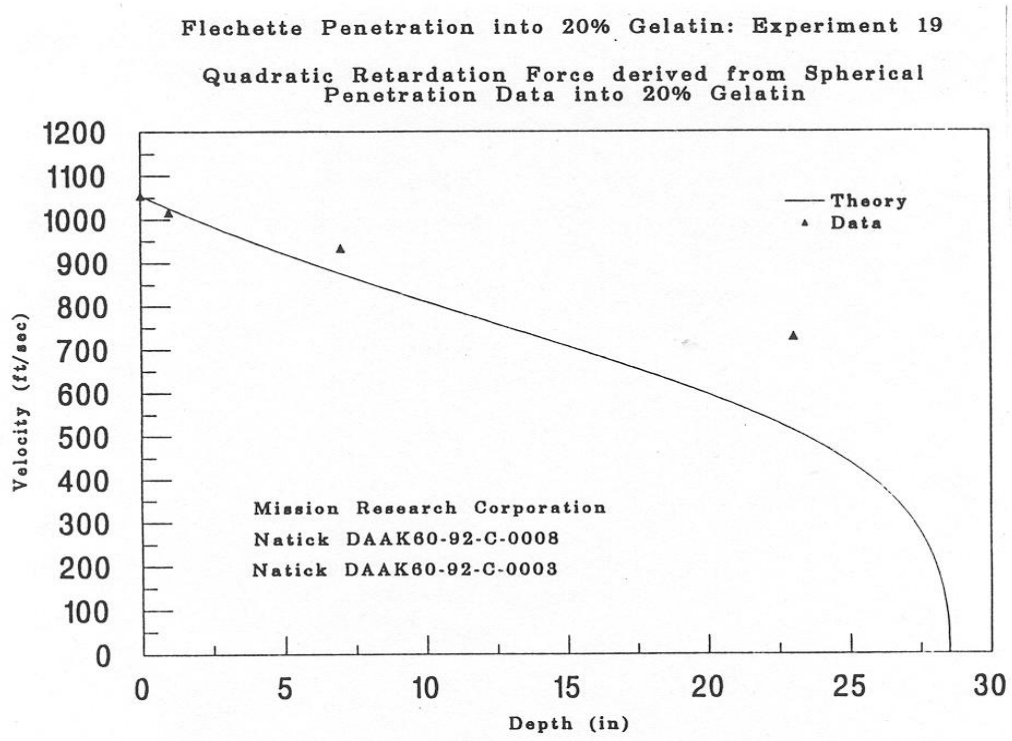

FIGURE 3-14. Flechette penetration into 20% gelatin: Theory and Experiment: Test 19

As concluding remarks to the above analysis, we would like to point out some important aspects of the theoretical modeling of penetration problems. First, in the absence of basic understandings of the underlying physics for low and high entry velocity penetration of a given target, it is not possible to model the nature the retardation force from experimental data. In this regard, equation (4) should be used as a guiding equation to connect the characters of the two fundamental function; $\delta(v)$, the penetration vs. entry velocity function and $F(v)$, the retardation force vs. local velocity function. Special care should be taken when we use the derivative of $\delta(v)$ to obtain $F(v)$, and large error may be introduced if proper functional forms [Equations (6) and (8)] are not used. Secondly, the usual quadratic form of the retardation forcing function $F(v)$ that is used in the **ComputerMan** Code, does not yield acceptable solutions when compared with the experimental data, and correct forms should be carefully incorporated in the analysis. It is interesting to mention here that a quadratic form is tempting because it predicts correct behaviors at both high and low velocity regions. But our problem here is to come up with a functional form that can be used for intermediate velocities as well. This is exactly where such functional form fails to provide the correct results. Since $\delta(v)$, depends on the integral of $F(v)$, error accumulation from the intermediate, local velocity zones are manifested in the incorrect prediction of the final penetration depth $\delta(v)$, for a given entry velocity v.

3.3 BODY ARMOR

3.3.1 Non-Penetrating Projectiles – Projectiles Impacting Body Armor

Extension of Vinson-Zukus FABRIC model for Armor Applications

Fabric armor made of Nylon, Kevlar etc. are increasingly used in designing lightweight armor materials. For proper designs of these armors, the ballistic characteristics of such materials need to be clearly understood. Some of the important ballistic characteristics include ballistic limits of fabrics used and their failure-to-strain data. In laboratory experiments under controlled environments, some features of the projectile penetration process have been observed. One of the most important of these features is the development of two distinct wave speeds in the fabric; one traveling with the sound wave speed of the fabric and other with an apparent transverse wave speed. On the arrival of the apparent transverse wave front, a discontinuity in particle velocity occurs indicating an abrupt change in acceleration. The observed physical phenomena can be explained as follows. A particle on the armor surface away from the impact location first sees the arrival of a tensile, longitudinal wave causing a particle motion towards the impact point. Assuming that the armor is initially on a horizontal plane, this horizontal particle motion ceases when the transverse wave arrives at the new displaced location of the particle. At this point in time, the particle now moves vertically along the direction of motion of the projectile with a velocity equal in magnitude to the instantaneous velocity of the projectile. If we assume that the magnitude of the tension is continuous at the point of such velocity discontinuity, we can establish a relationship between the jump in velocity and the tension of the fabric at that point

through the equation of motion of the particle. Unlike a solid material where a bending can be realized, these wave speeds are also dependent on the instantaneous strain of the fabric. The deformed shape of the fabric at any given time closely approximates a cone as shown in Figure 3-15.

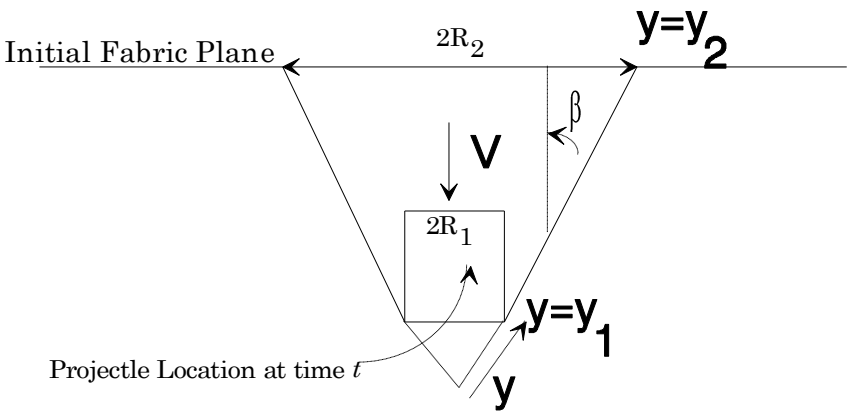

FIGURE-3-15. Deformed Shape of a Fabric due to Projectile Impact

Analytical Model

To describe a mathematical model of the above process, we introduce the following notations that identify the governing physical variables at any time *t*.

c = Sound wave speed in the fabric
E = Elastic modulus of the fabric
T = Tension in the fabric
ε = Fabric strain
M = Mass per unit length of fabric yarn
σ = Fabric stress
ρ = Fabric density
u = Transverse wave speed in the fabric
$\bar{u}$ = Apparent transverse wave speed in the fabric
V = Instantaneous projectile velocity
w = Particle velocity

Using the laws of mathematical physics and the definitions of these variables, we can establish the following relations:

$$c = \sqrt{\frac{E}{\rho}} = \sqrt{\frac{(\frac{\partial T}{\partial \varepsilon})_{\varepsilon=0}}{M}}$$

$$u = \sqrt{\frac{\sigma}{\rho(1+\varepsilon)}} = \sqrt{\frac{T}{M(1+\varepsilon)}}$$

$$\bar{u} = (1+\varepsilon)u - w$$

$$w = c\varepsilon$$

$$T = \sigma A = EA\varepsilon = \rho c^2 A\varepsilon = Mc^2\varepsilon \qquad (36)$$

$$u = c\sqrt{\frac{\varepsilon}{1+\varepsilon}}$$

$$\bar{u} = c[\sqrt{\varepsilon(1+\varepsilon)} - \varepsilon]$$

$$V = \sqrt{(1+\varepsilon)^2 u^2 - \bar{u}^2} = c\sqrt{2\sqrt{\varepsilon^3(1+\varepsilon)} - \varepsilon^2}$$

$$\varepsilon = \frac{V^2}{4c^2} \text{ , for small } \varepsilon$$

Besides these relations, the stresses and the displacements of the fabric at any point inside the cone can be calculated from the equations of motion (which are the equations of equilibrium if the fabric mass in an elementary area is considered small with respect to the mass of the projectile) and stress-strain-displacement relations. With reference to Figure 3-16, the equations of motion of an elementary area along *y* yields the following equations:

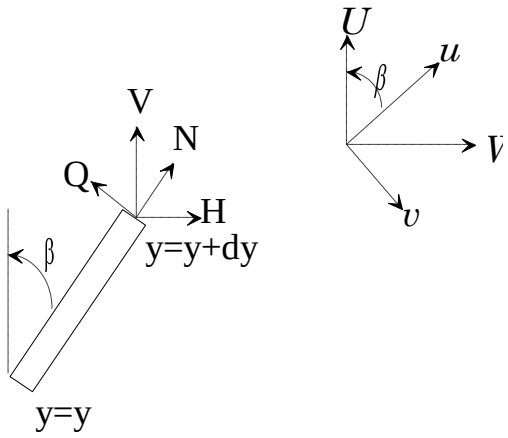

FIGURE-3-16. Stress-strain-displacement relations: Equations of Motion

Kinematic Constraints

$$W = 0 = Q \qquad (37a)$$

Force Relations

$$N = V\cos\beta + H\sin\beta$$
$$Q = -V\sin\beta + H\cos\beta \tag{37b}$$

Displacement Relations

$$u = U\cos\beta + W\sin\beta$$
$$w = -U\sin\beta + W\cos\beta \tag{37c}$$

Stress-strain-displacement Relations

$$E\varepsilon_y = E\frac{\partial u}{\partial y} = \frac{N}{h} \tag{37d}$$

Equation of Motion

$$V(y) = \frac{R_1}{R}V(y_1) = \frac{\Theta}{R} \text{ where } \Theta = V(y_1)R_1 \tag{37e}$$

where Θ is the only loading term which includes the vertical retarding force $V(y_1)$ on the projectile. In (37), h is the thickness of the fabric.

From (37), we get the displacement $U_p = U(y = y_1)$ of the projectile as

$$U_p = \frac{\Theta \ln\left(\frac{y_1}{y_2}\right)}{Eh\sin\beta\cos^2\beta} \tag{38}$$

Strain can be obtained from (37e) as

$$\varepsilon = \frac{U_p \sin\beta\cos\beta}{R_1 \ln\left(\frac{y_2}{y_1}\right)} \tag{30}$$

Equations 36 through 39 can be used to describe the motion of the projectile as a function of time when the properties of the fabric are given. We come back to the discussions of the methods for solving these problems. Next we give a brief description of a numerical algorithm that has been developed to solve the above problem using a system of connected masses and springs.

Numerical Solution

One of the fundamental differences between elastic (and/or composite) and fabric material is the respective presence and absence of bending modes in the deformed shapes. Thus it is generally believed that a fabric can be modeled as a planar collection of mass particles that are connected by *massless* elastic strings. The mass of each of these particles needs to be carefully calculated so that local kinematics and the differential equations of motion can dictate the actual fabric behaviors under any loading condition. Usually, this is done by dividing the fabric in strips and using a uniform distribution of the total mass along equally spaced nodes along a strip. Since crimping is one of the key parameters affecting the mass distribution in a fabric, such particle mass distribution includes a properly defined *crimp* factor. A numerical code called the FABRIC code incorporates one such discrete mass/string model where the numerical calculations are performed according to the following algorithms:

1. When we input the number of nodes along the length or width of the fabric panel, a stability condition is used to calculate appropriate time interval ensuring that during each time interval only neighboring nodes on the fabric are set in motion.

2. To initiate the motion, it is assumed that during the first time interval, only the modes below the projectile footprint, are set in motion with velocity equal to the impact velocity of the projectile.

3. Strains in the fabric are then calculated from the deformed shape of the fabric, which in turn gives the tension along the neighboring nodes of the projectile. Application of Newton's second law of motion gives the deceleration of the projectile and the acceleration of the neighboring nodes. This process is repeated during the next time interval and is continued until the projectile either stops or the fabric fails to provide any more retardation to the projectile.

The existing version of the analytical fabric code was developed by Vinson and Zukus in 1975[33]. The basic analytical model and associated equations are described earlier. For the solution of a particular problem, the method used by Vinson and Zukus assumes that the history $\bar{u}(t)$ of the transverse wave velocity is known from experimental data. This is a very strict constraint on the usability of the model since it is very difficult to generate these data. Besides, since the distribution of $\bar{u}(t)$ is dependent on the strains and the impact scenario, such data is useless for other impact conditions even when the same fabric panel is used. We modified these requirements so that fabric panel properties independent of the impact conditions, can be used as input to the code. There is also a serious error in the calculation procedure presented in their paper, which has also been corrected in our model. We first present the method used in the Vinson and Zukus model and then we discuss our modifications and corrections. For the execution of these steps, we assume that a) the transverse wave speed $\bar{u}(t)$, b) the initial projectile velocity v_p and c) the projectile radius R_1 are known.

[33] Vinson J. R., and Zukus, J. A., "On the Ballistic Impact of Textile Body Armor," *Journal of Applied Mechanics*, 263-268, June, 1975

1. Select a time interval Δt to use.
2. From known $\bar{u}(t)$ *vs*. *t*, calculate outer cone radius R_2 (Figure 3-15) using

$$R_2 = R_1 + \frac{\partial \bar{u}}{\partial t}\Delta t$$

3. Calculate the projectile displacement U_p from $U_p = v_p \Delta t$

4. Calculate semi-angle β of the deformed cone from $\beta = \tan^{-1}\left(\frac{R_2}{U_p}\right)$

5. Calculate strain from equation (28).

6. Using equations (25)-(28), Young's modulus *E* , sound wave speed *C* and the projectile retardation force *V* are given by

$$C = \sqrt{\frac{E}{\rho}} = \frac{\bar{u}}{\sqrt{\varepsilon(1+\varepsilon)} - \varepsilon}$$

$$V = \left(\frac{Eh}{R_1}\right)\frac{U_p \sin\beta \cos^2\beta}{\ln\left(\frac{R_2}{R_1}\right)}$$

7. Projectile deceleration d_c is given by the equation of motion $d_c = \frac{2\pi R_1 V}{m_p}$.

8. New projectile velocity, displacement and R_2 are now calculated from

$$v_{p\ t+\Delta t} = v_{p\ t} - d_c \Delta t$$

$$U_{p\ t+\Delta t} = U_{p\ t} + v_p \Delta t - \frac{1}{2} d_c\ \Delta t^{\ 2}$$

$$R_{2\ t+\Delta t} = R_{2\ t} + \frac{\partial \bar{u}}{\partial t}\Delta t$$

9. Steps 4-9 are repeated until either the projectile stops or the fabric suffers a strain failure.

Modifications

The steps 1-9 mentioned above are modified to make the existing analytical model more usable and yield better and correct results. First, the expression for the semi-angle β of the deformed cone used by Vinson and Zukus and given in step 4 above, is incorrect. The correct expression for β is

$$\beta = \tan^{-1}\left(\frac{\frac{\partial \bar{u}}{\partial t}\Delta t}{u_p}\right)$$

Next we change the solution procedure to bypass the requirement for prior information of $\bar{u}(t)$ vs. t history. It is relatively easier to get the fabric property through its E vs ε data. For a given fabric panel such data be obtained by simple laboratory experiments, and these data can be used under any impact conditions. Accordingly, we have made the following changes to the calculation algorithm of existing analytical model, and we refer to this model as the *corrected analytical model.* In using these steps, it is assumed that for the fabric under consideration E vs ε data is known *a priori*.

1c. At the start of the penetration process i.e., for small strain ε, known Young's modulus *E* is used to calculate the sound wave speed *c*. Since the impact velocity V is known, we can use the last equation in (1) to calculate the strain from

$$\varepsilon = \frac{V^2}{4c^2}$$

2c. From known ε, *E* can be calculated which yield *c* from $c = \sqrt{\left(\frac{E}{\rho}\right)}$. From equation (1), the apparent transverse wave speed is known as a function of strain; $\bar{u}$ is then calculated from equation (1).

3c. Rest of the steps is similar to the steps (1-9) of the existing analytical model except that c is calculated from *E* instead of *E* being calculated from c.

An Example

To evaluate the merits, demerits and applicability of the modified fabric code for applications in armor penetration, existing analytical code and the modified analytical code, we have selected the same example that Vinson and Zukus used in their paper mentioned previously. In this problem, a 17 grain 22 caliber projectile impacts a Nylon fiber panel. The properties of the target panel and the impact conditions are given below;

Projectile mass:	1.1042 *gm*
Projectile footprint radius:	0.1075 *in*.
Initial projectile velocity:	7480 *in/sec*
Fabric thickness:	0.0163 in.
Textile density:	1.15 gm/cc

E vs. ε data: Estimated from Vinson & Zukus paper

We have some difficulty in estimating E vs. ε data from the Vinson and Zukus paper since it yields two values (Figure 3-17) of E from the same ε. We only used the first single valued branch of the data and then extrapolated the results outside the range. E vs ε curves for the Vinson and Zukus model and the corrected analytical model are shown in Figure 3-17.

Results for the projectile velocity history obtained for this case from the three methods discussed above are graphically presented in Figure 3-18. In Figure 3-19, we have included two results from a numerical code called the FABRIC code by changing the number of nodes used along the fabric panel, and the experimental data for comparison. These results seem to converge and no appreciable changes are observed by increasing the nodes even further.

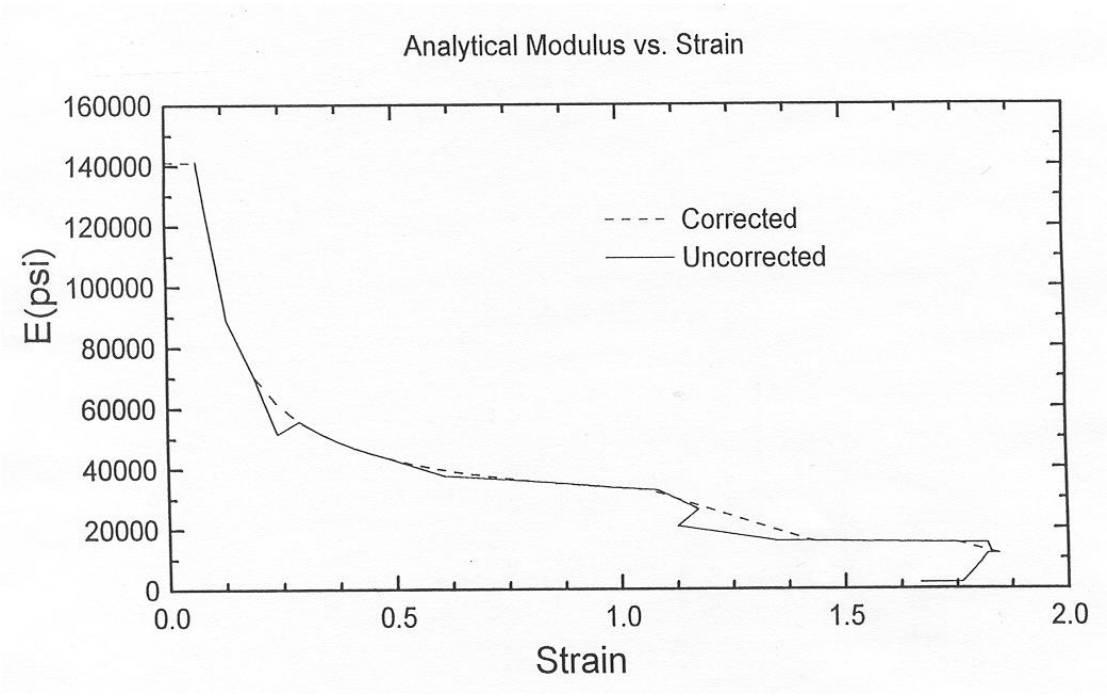

FIGURE 3-17. KEVLAR Fabric: Elastic Modulus vs. Strain Curves

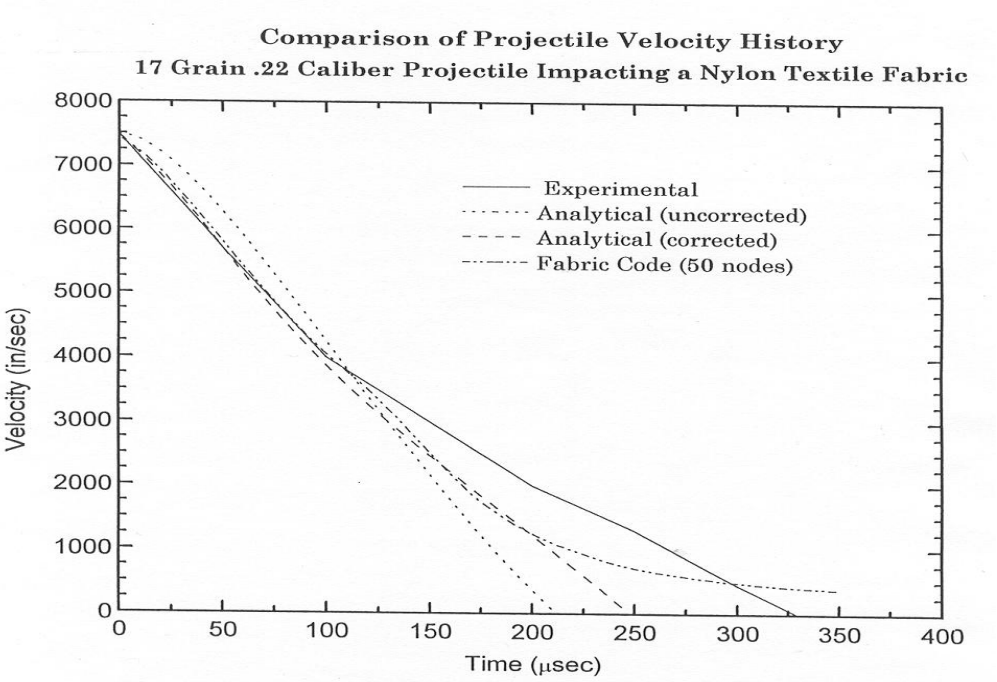

FIGURE 3-18. Comparison of Projectile Velocity Histories: Various Methods

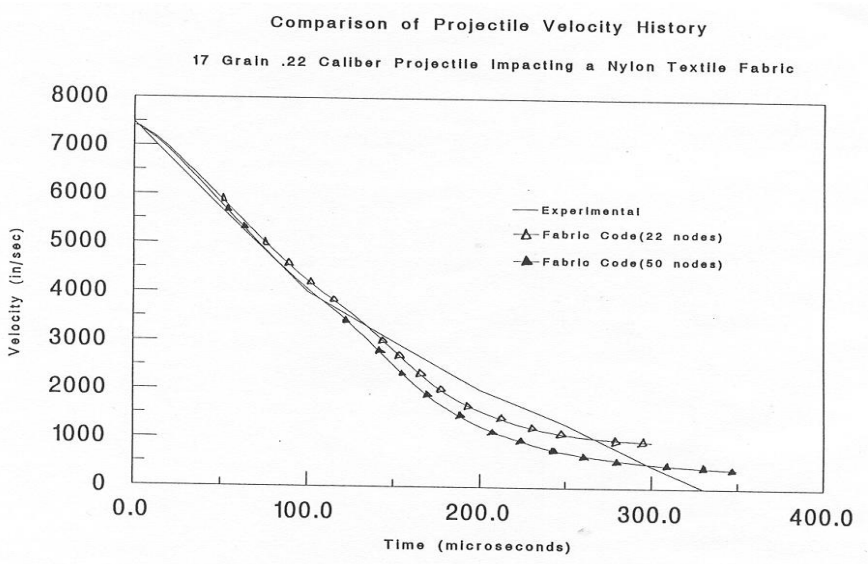

FIGURE 3-19. Comparison of Velocity Histories

Modified Projectile Retardation Model: Retardation force on projectiles impacting a single or multiple layers of textile materials.

Equations of motion governing the projectile history involve forces of retardation from the fabric surrounding the projectile at any given instant of time. These forces are related to the local strains

in the immediate vicinity of the projectile contact area with the impacting fabric material. For most fabrics, the stress-strain relation is nonlinear and hence a closed form solution in the hodograph domain obtained by integrating the equations of motion in the velocity-space domain is not usually feasible.

From the experimental data available to us from tests on various types of Kevlar materials with or without 20% gelatin backing, some preliminary data on the force of retardation can be obtained from simplified equations of motion governing these projectiles. For example, let us consider a projectile of mass *m* (treated as a particle mass) impacting a layer of fabric material with an incident velocity of v_0, travels a distance *d* prior to its arrest. The mean force of retardation $\bar{f}$ acting on the projectile can be estimated from the equation of motion of the projectile under a constant retardation force $\bar{f}$ equal to the mean force of retardation over the period of penetration t_s. These give

$$\bar{f} = \frac{mv_0^2}{2d} \tag{40}$$

For a mean radius *r* of the projectile contact area with the fabric of thickness *h*, mean strain $\bar{\varepsilon}$ can be calculated from

$$\bar{\varepsilon} = \frac{\bar{f}}{E(2\pi rh)} \tag{41}$$

where *E* is the mean elastic modulus of the fabric under impact.

Application of the above analysis to the experimental data, provides us with some important information regarding the strains developed in fabric during the projectile penetration and the governing mechanism of the failure of fabric materials. In what follows, we present some calculations based on the above analysis as well as from some analytical models on impacting fabrics so that a comparative analysis can be made.

Both Vinson-Zukus analytical model and available computational models predict unacceptable high strain in the immediate vicinity of the projectile. To avoid these high strains, we use a different method to solve the problem of projectile penetration of fabric materials. With reference to Figure 3-21, since both the deformed shape and the undeformed state of the fabric is known, we may calculate the average strain from equation (32). But since the strain is zero at A which is the boundary between the disturbed and undisturbed zone, and maximum at the projectile footprint, we redistribute the strain using some distribution conforming to these constraints. Results show that a linear fit does not provide enough strain and hence enough retardation force to stop the projectile at levels observed in experiments. The reason for the failure of the linear fit model is the local enhancement (Figures 3-20 and 3-21) of the strain in the immediate vicinity of

the projectile due to the large relative displacement of the projectile with respect to the fabric. This has been observed in the photography of the deformed shape of the projectile during experiments. To include this effect in our calculations we have included a local enhancement factor over the linear model. For the penetration data of unsupported KEVLAR, this non-dimensional factor is of the order of 4-5. We hope that this factor will be material independent so as to allow us to use the revised models for prediction of penetration of unknown fabric or for fabric for which no experimental data is available. More experimental data from other fabrics is necessary to confirm this hypothesis. The revised computational steps which completely bypasses the analytical formulae of Vinson-Zukus results are as follows:

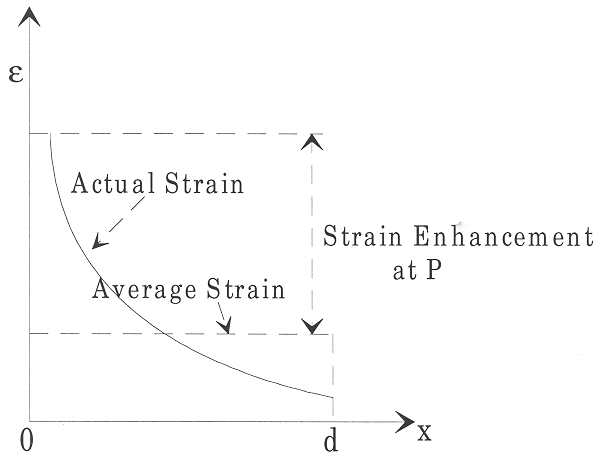

FIGURE 3-20. Local Strain Enhancement at the Projectile-Target Footprint

For a given time step, we first calculate the average strain from the deformed shape of the fabric using the equations given in Equations 40 and equation 41. The non-uniform strain distribution is the obtained using the local enhancement factor over the linear fit model. This gives us the strain ε at the projectile footprint. For the E vs. ε (Young's modulus versus strain) data on the fabric, we calculate E and then the force of retardation from the product $E\varepsilon$. This force then gives the velocity of the projectile for the next time step. For the 6-layer KEVLAR-29, computational results agree well with the experimental data.

In the new formulation, we have modified the deformation geometry of Vinson-Zukus model to accommodate the contoured bottom of the projectile. In the Vinson-Zukus model, the projectile is assumed to have a flat bottom so that the conical angle is clearly defined with a known base which is the same as the projectile footprint. In the case of a contoured bottom, the projectile base in contact with the fabric at a given instant of time should be calculated from the condition of tangency of the fabric line from the contact point between the projectile and the fabric (Figure 3-21).

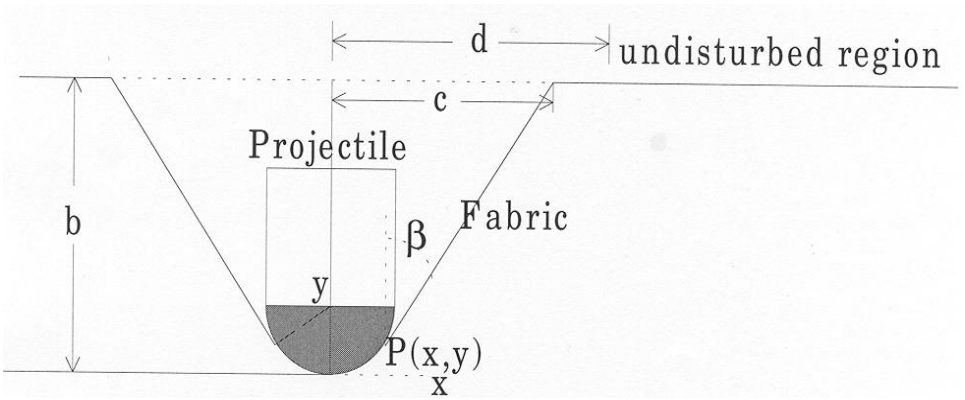

FIGURE 3-21. Deformation of a Fabric due to a Projectile with Contoured Bottom

In the following equations, we assume that the projectile bottom is a semicircular area of radius *a*. Then the equations that relate the radius of the projectile contact area with the fabric, to the other geometrical parameters of the deformed shape (Figure 3-21) of the fabric, are as follows:

Projectile Bottom:

$$y = a - \sqrt{a^2 - x^2}$$

Tangency condition at $P(x, y)$

$$\frac{b - a + \sqrt{a^2 - x^2}}{c - x} = y'(x) = \frac{x}{\sqrt{a^2 - x^2}}$$

This gives

$$x = \frac{ca^2 \pm a(b-a)\sqrt{c^2 + b^2 - 2ab}}{c^2 + (b-a)^2}$$

However, due to the enhanced strain in the vicinity of the projectile footprint, the direction of the stress-induced retardation force is mostly vertical so that we have modified the equations of motion of the contoured projectile accordingly. These changes are incorporated in the calculations for comparing the results with the experimental data. For KEVLAR 29, we have some experimental data that we used to estimate the range of validity of our approach. For KEVLAR 29 and other types of fabrics (e.g., Generic PVA and KEVLAR 149), results show our estimates based on the revised strain model described above.

3.3.2 Penetrating Wounds

MATHEMATICAL MODEL OF STEEL CUBES PENETRATING ARMORS

Empirical equations have been derived from experimental studies on steel cubes penetrating armors.[34] These equations can be used to determine the critical velocity of penetrations and residual velocity for various armors. These equations are given below, and have been used here in determining penetrating wounds in personnel wearing armors. The definitions and units of various physical and empirical quantities used in the model are as follows:

υr = Residual velocity after armor penetration in $\frac{m}{\sec}$

υs = Striking velocity of the cube at a given obliquity θ in $\frac{m}{\sec}$

υc = Threshold velocity for penetration at a given obliquity angle θ in $\frac{m}{\sec}$

$B_{arm}, N_{arm}, K_{arm}, K1_{arm}, K2_{arm}$ = Components of armor-dependent, material properties matrices where the first two matrices are in consistent units while others are non-dimensional.

$K2 = 155\frac{gm}{cm^2}$; L = Cube length in centimeters (cm)

arm = Armor identifying integer lying between 1 and 9. These identifications are as follows. arm = 1 (Army Nylon Vest), 2 (Marine Doron Vest), 3 (Titanium Vest), 4 (Nylon Helmet), 5 (Nylon Liner), 6 (Polycarbonate Helmet), 7 (Steel Helmet), 8 (Titanium) ,1 (PC Helmet), 9 (Titanium Type III Helmet).

The experimental values of these matrices are

[34] "Expected residual velocities of cubes perforating armor material as a function of projectile size and obliquity angle", BRL Report No. 2011, September, 1969.

$$B = \begin{pmatrix} 1 \\ 2.21 \\ 2.28 \\ 1.38 \\ 1.88 \\ 2.3 \\ 2.41 \\ 3.15 \\ 0.31 \end{pmatrix} \quad N = \begin{pmatrix} -0.053 \\ -0.163 \\ -0.167 \\ -0.089 \\ -0.131 \\ -0.17 \\ -0.169 \\ -0.21 \\ 0.12 \end{pmatrix} \quad K = \begin{pmatrix} 1.61 \\ 1.56 \\ 1.65 \\ 1.47 \\ 1.68 \\ 1.77 \\ 1.84 \\ 1.47 \\ 1.99 \end{pmatrix} \quad K1 = \begin{pmatrix} 4.34 \\ 4.86 \\ 4.64 \\ 4.71 \\ 4.69 \\ 4.71 \\ 5.1 \\ 4.62 \\ 4.38 \end{pmatrix} \quad K3 = \begin{pmatrix} 0.145 \\ 0.127 \\ 0.148 \\ 0.119 \\ 0.091 \\ 0.122 \\ 0.135 \\ 0.155 \\ 0.206 \end{pmatrix}$$

A = Projected area of the cube on the armor surface; ρ = Cube density in $\frac{gm}{cm^3}$

$\theta x, \theta y$ = Cube rotation about cube x- and y-axis, respectively (see Figure 3-20)

θ = Obliquity angle of the striking velocity with the normal to the armor surface (see Figure 3-22).

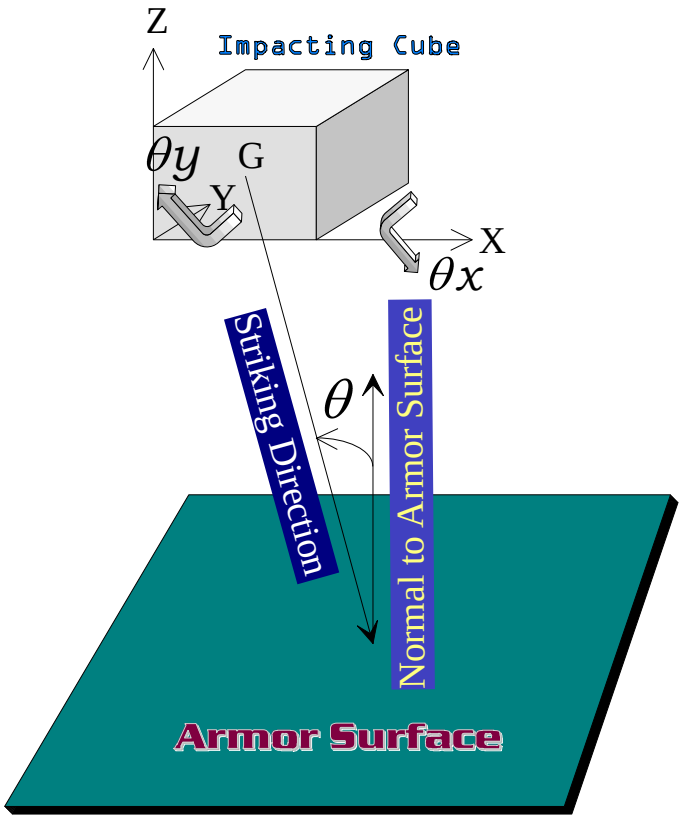

FIGURE 3-22. Schematics of a Cube Projectile Impacting an Armor

EMPIRICAL EQUATIONS RELATING MODEL VARIABLES

Projected area $A = L^2 \sum_{i=1}^{6} \cos\ \pi - \theta_i \left[H\left(\frac{\pi}{2} - \theta_i\right) - H\ \pi - \theta_i \right]$ where

$\theta_1 = \cos^{-1}\ \cos\theta x \sin\theta y\ , \theta_2 = \pi - \theta_1, \theta_3 = \pi - \theta x, \theta_4 = \theta x, \theta_5 = \cos^{-1}\ \cos\theta x \cos\theta y\ , \theta_6 = \pi - \theta_5$

$H\ \theta$ is the Heaviside unit step function.

$$\upsilon c0(arm) = e^{\left[K1_{arm}\left(\frac{A}{m}K2\right)^{K3_{arm}} \right]} - 1$$

$$\upsilon c(arm) = K_{arm}^{\sec\theta - 1}\, \upsilon c0(arm)$$

$$\upsilon r(arm) = \upsilon s - \upsilon c(arm) \left(e^{\left[B_{arm}\ \upsilon s - \upsilon c(arm)\ ^{N_{arm}} \right]} - 1 \right)$$

For example, if we use army nylon vest ($arm = 1$), $\upsilon s = 928 \frac{m}{\sec}, \theta = 10^{\circ}$ deg. , we get the residual velocity $\upsilon r = 523 \frac{m}{\sec}$.

Another example is shown in Figure 3-23 where we compared the residual velocities for different armors for various obliquity angles at a striking velocity of $928 \frac{m}{\sec}$.

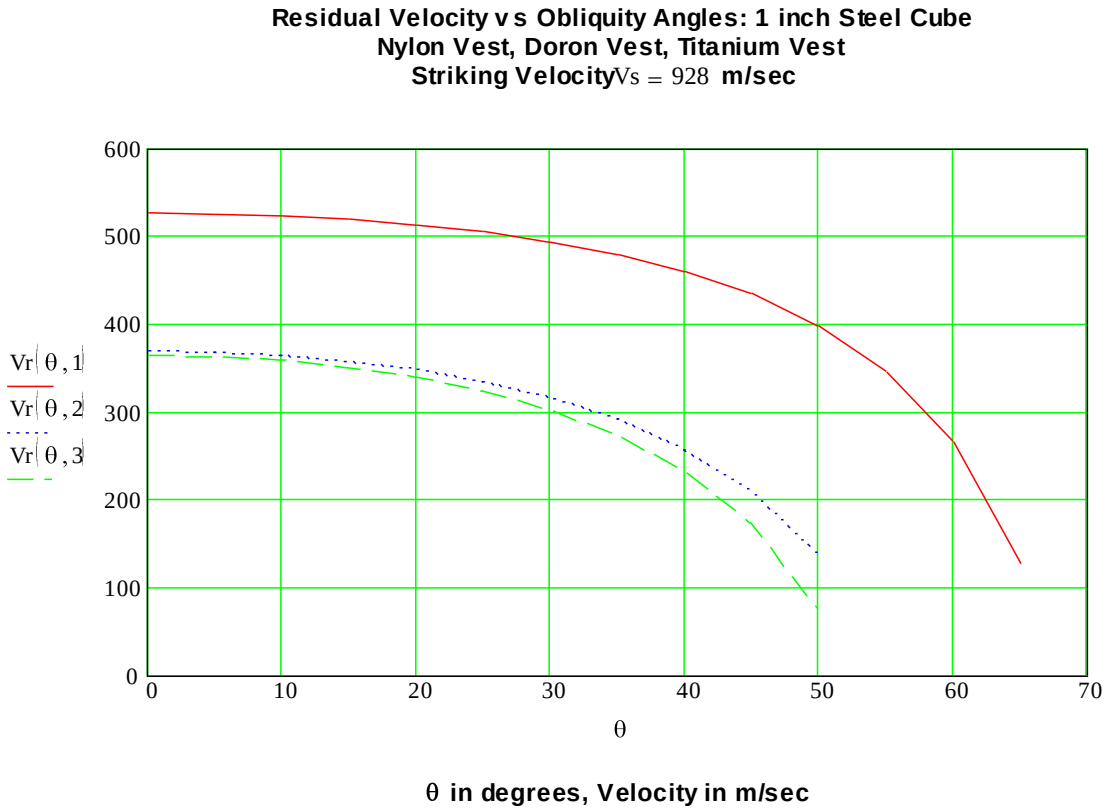

FIGURE 3-23. Residual velocity for various armors at different obliquity angles

Using the above analysis, we can calculate the residual velocity due to given fragment impacting an armor. For other projectile geometry, we use the same basic functional relationships except that the projected area, mass and density will change accordingly. When the armor is defeated, the residual velocity is then passed onto the MRCMAN code for determining the wound trajectory and associated damage analysis.

BLUNT TRAUMA

When the striking velocity of a projectile is less than the a critical threshold velocity, the projectile does not penetrate a given armor but a person standing nearby may be injured due to blunt trauma. To analyze this scenario, we use the body armor analysis described in the previous sections where the body underneath the armor is modeled as an elastic foundation. At the present time, since no reliable blunt trauma criterion is available, we use the deformation of the body underneath the impact location as a measure of the severity of the blunt trauma.

3.4 BLAST

3.4.1 Direct Blast Injuries

The assessment of direct blast injuries is based on empirical environment data specified in terms of peak pressure and duration.

3.4.2 Secondary Blast Injuries

Secondary blast injuries result from grenade or other building material fragments (glass window etc.) that are ejected when an explosion occurs. The cases of grenade and glass fragments impacting a person are analyzed below.

Grenade Fragments: Distribution of Mass and Velocity

The type of grenade used in this study is a M-406 high explosive grenade with steel casing. Thus the grenade fragments are steel fragments, and their mass and velocity distributions over polar angles (Figure 3-24) are given in Figures 3-25 and 3-16, respectively. Following assumptions are made in describing the fragment kinematics. It is assumed that all fragments emanate from the grenade center of mass G (Figure 3-24), and they travel to a field point P along a straight trajectory. If x is the horizontal distance between G and P, then the velocity at P of a fragment of mass m and fragment projected area A is related to the initial ejection velocity v_0 through the empirical equations

$$\frac{v}{v_0} = \frac{y}{e^y - 1}\ , \ y \neq 0$$

$$y = \frac{x}{\left(\frac{2m}{C_D \rho A}\right)}$$

where y is the non-dimensional distance, ρ is the fragment density and C_D is the drag coefficient. The fragment projected area A plays a very important role in retarding the fragment since it controls retardation force. It is defined as the projection of the fragment surface on a plane normal to the velocity of the fragment. It is empirically related to the fragment mass as $A = bm^{c+1}$ where b and c are curve fit parameters of the experimental data. For applications in secondary blast injuries, fragment mass, projected area and velocity are input to the MRCMAN code for penetrating wound analysis or blunt trauma analysis using related codes.

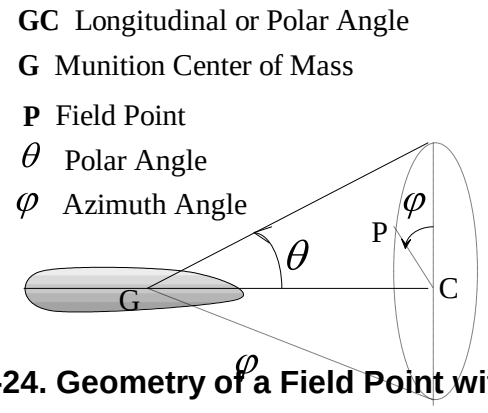

FIGURE 3-24. Geometry of a Field Point with respect to a Grenade

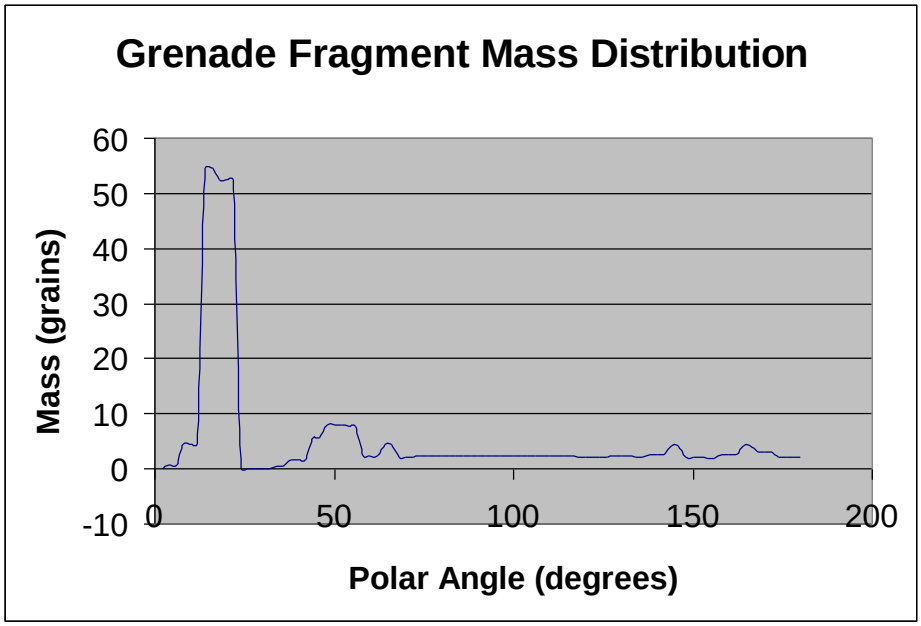

FIGURE 3-25. Mass Distribution vs. Polar Angle from Grenade Blast

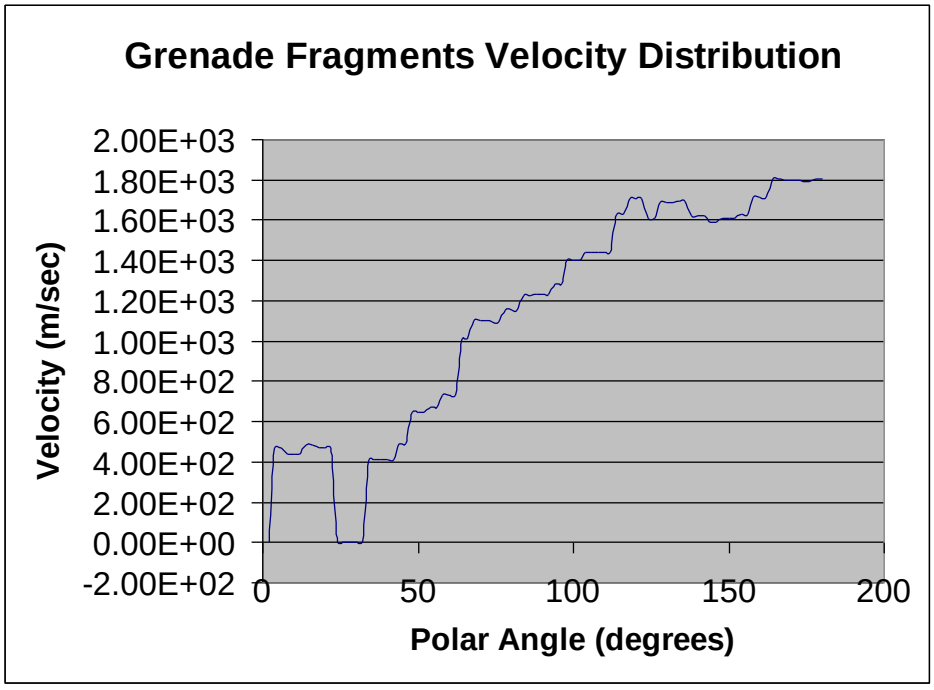

FIGURE 3-26. Velocity distribution vs. polar angles from Grenade Blast

Secondary Blast Injuries from Window Glass Fragments

Some experimental data and related empirical models are available to determine the properties of window glass fragment and related kinematics due to blast wave's incident on window surfaces from grenade explosions. These fragments may move with significant velocities through air and may cause considerable damage upon impact with human bodies. For the determination of the associated injuries, we need to establish equations similar to those of the grenade fragments discussed earlier. Basic equations of the glass fragment model from blast exposure are given below. The mean velocity, mass and fragment area vs. blast overpressure are shown in Figures 3-25 through 3-29.

Model Parameter Definitions and Units:
M = Mass in grams
V = Velocity in m/sec
P = Overpressure in KPa
ρ = Desnity in gram/cc
t = Thickness in cm
A = Area in cm**2

Input:
Thickness $t := 0.516$
Density $\rho := 2.5$

Mean Mass vs. Mean Velocity Calculation

$V(M) := e^{3.132-0.255\ln(M)}$

$V(5) = 15.205$

$M(P) := t\cdot\rho\cdot e^{4.2643-\sqrt{12.5+0.00345P^2}}$

$A(P) := \frac{M(P)}{t\cdot\rho}$

$tst := V(M(6))$ $M(6) = 2.627$

$tst = 17.916$ $V(2.627) = 17.916$

$M(100) = 0.097$

$P := 1,2..100$

$M(3) = 2.662$

$M(10) = 2.547$

$V(M(10)) = 18.058$

$V(M(3)) = 17.856$

$M(60) = 0.623$

$V(M(60)) = 25.858$

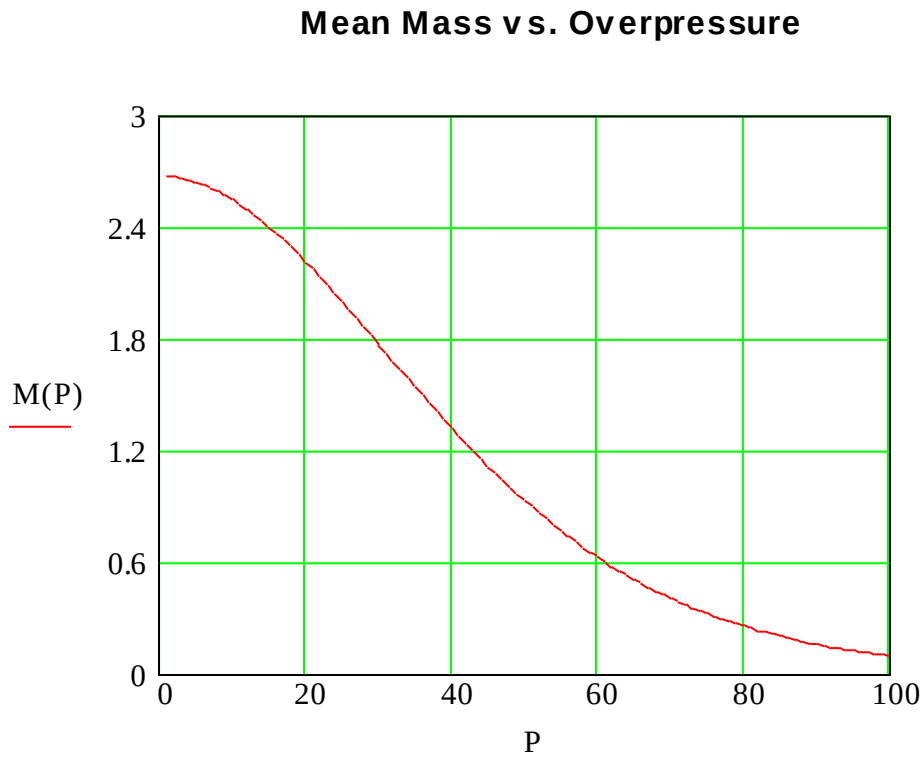

FIGURE 3-27. Distribution of Mean Glass Fragments vs. Blast Overpressure

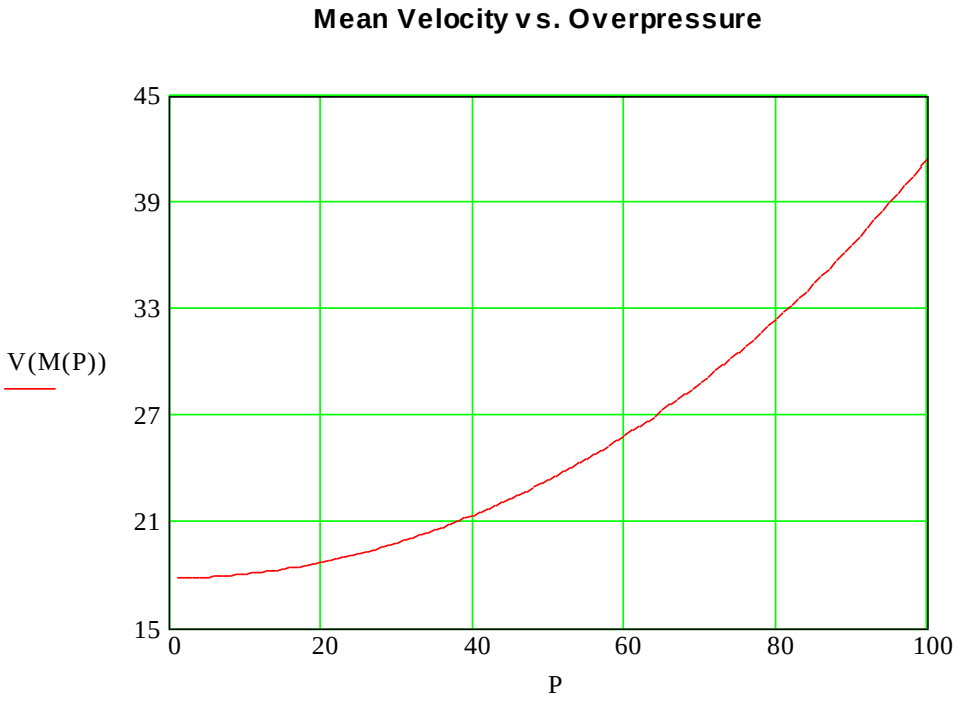

FIGURE 3-28. Average Velocity of Glass Fragments vs. Blast Overpressure

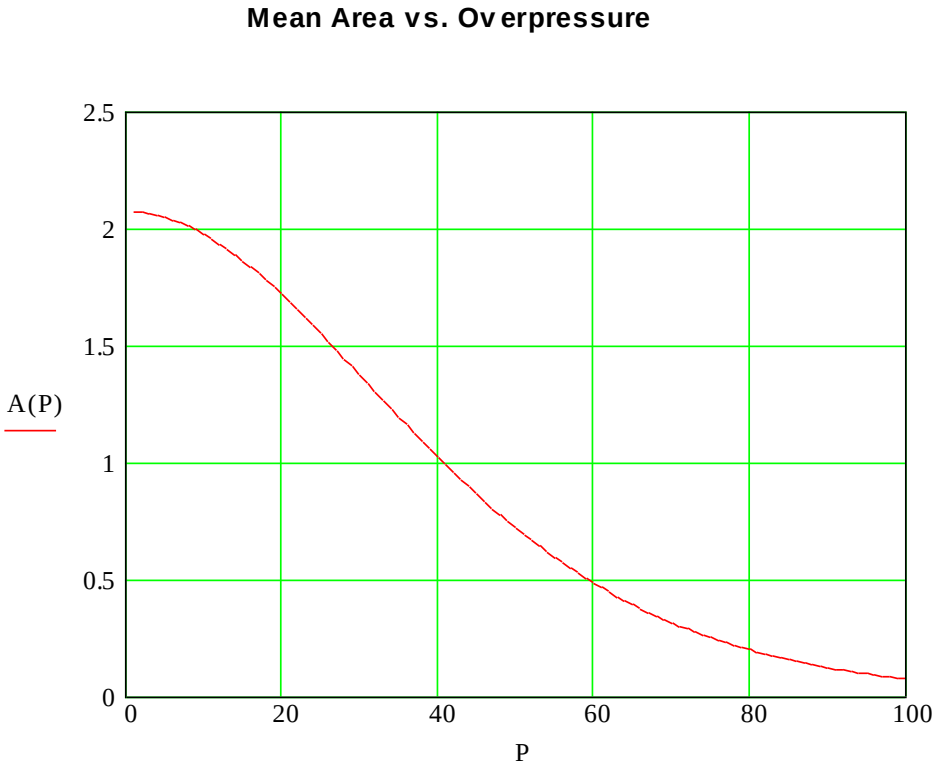

FIGURE 3-29. Distribution of Average Glass Fragments Area vs. Blast Overpressures

3.4.3 Tertiary Blast Injuries

Tertiary blast injuries results from kinematic trauma of the human body colliding with an object. This is estimated by taking the free field environment an allocating impulse to the various joints on the human body using ATBM (recall from Sections 1 and 2 that ATBM input files are created as part of MRCMan execution when blast environments are possible). The motion of the human body is then calculated using ATBM for the duration of the blast produced impulse on the human body. The resulting ATBM calculated motion and joint articulation is used to determine body collision parameters. These collision parameters are then compared with blunt trauma thresholds (currently the Viano viscous criteria) and bone fracture criteria determined earlier in terms of energy density per unit time.

4. CHARACTER SIMULATOR SOFTWARE DEVELOPMENT (MRCMAN32 Program)

4.1 MAIN MENU DIALOG

MRCMAN32 main menu items include Select Projectile, Autopsy Mode, Link Mode, Projection Mode and Visible Human Database Processing. Under Select Projectile, six different types of projectiles are available which include bullet, cube, cylinder, flechette, fragment or sphere. Autopsy Mode is a re-creation of the original Computer Man ballistic analysis program, and is based on the Eyclechymer and Shoemaker tissue database. The Autopsy Mode man is a full body model and can not be articulated. Under Link Mode the man is represented as an articulating link and tissue model, segmented into individual appendages. Either the Visible Human or Eyclechymer and Shoemaker tissue database may be used for ballistic analysis in Link Mode. Under Visible Human Database Processing the user may generate ATB and ISM input files, run the ISM Internal Structure Modeler or run Scanfit which maps Visible Human data to body surface scans.

Figure 4-1. MRCMAN32 Main Menu Screen

4.2 PROJECTILE MENU

From the Select Projectile menu, six different types projectiles may be chosen for use in ballistic analysis. Available projectiles include bullet, cube, cylinder, flechette, fragment, and sphere. The "current projectile" is the projectile chosen by the user for ballistic analysis. The current projectile is set for both Autopsy Mode and Link Mode, by selecting a projectile from the Select Projectile sub menu, and then clicking the OK button in the sub menu dialog. When starting the MRCMAN32 program, the current projectile is the same as it was when the program was last terminated. If the user exits MRCMAN32 with flechette selected as the current projectile, then flechette will be the current projectile the next time MRCMAN32 is started.

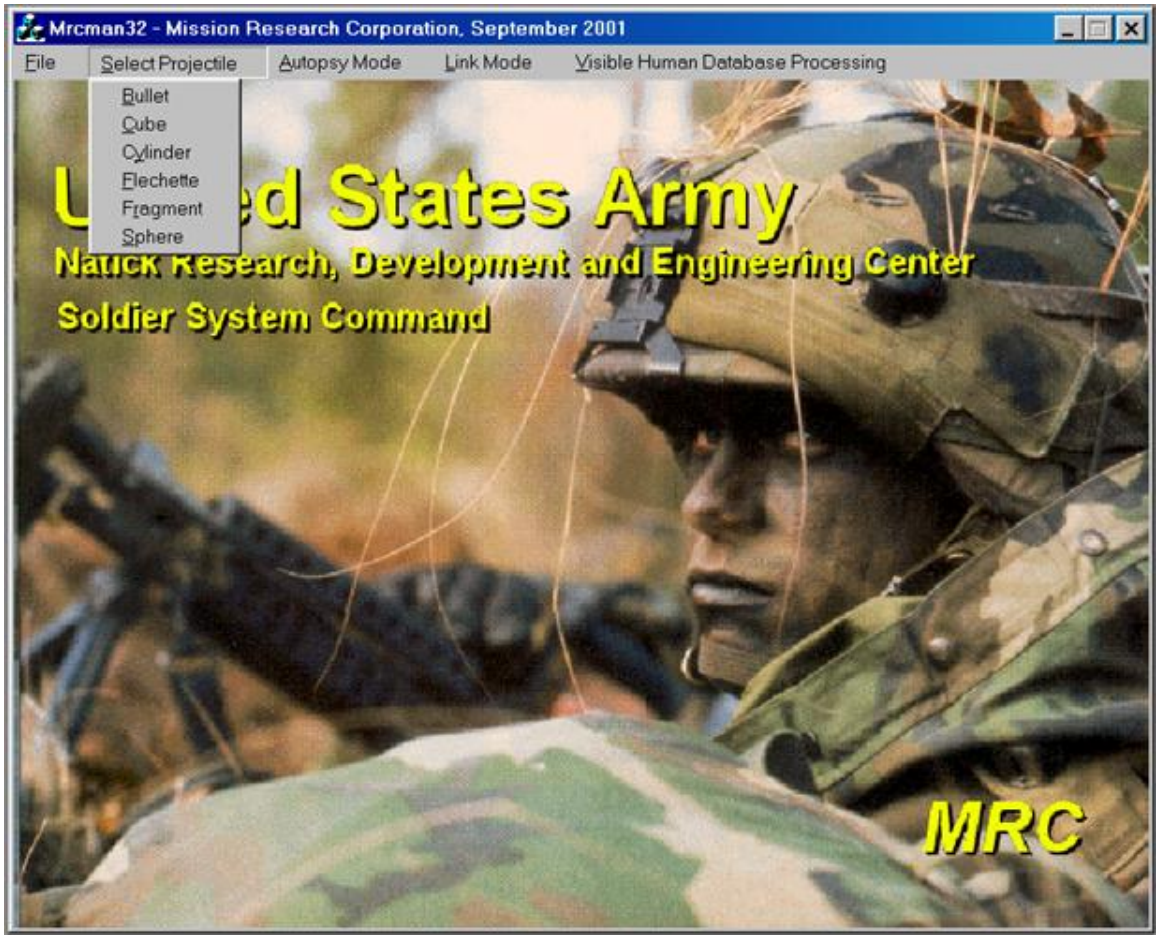

Figure 4-2 – Menu of Available Projectiles

Select Projectile Sub Menu Dialogs

Each projectile listed in the Select Projectile sub menu has its own sub menu dialog. The bullet sub menu dialog is presented as an example below in Figure 4-3. A schematic diagram describing the projectile variables is in the left upper corner of the dialog. Below the diagram is a photograph depicting a typical projectile of that type. In the upper right hand corner are edit

boxes, where projectile geometry and density variables may be adjusted. Geometry and density data are used to compute the current projectiles mass, which is used in Autopsy Mode and Link Mode ballistic calculations. In the case of a fragment, projectile mass is entered directly.

If the user presses the OK button, the current projectile is set to the projectile represented by the dialog, and current projectile mass is computed from the geometry and density data represented in the edit boxes. The Apply button simply forces the update of the edit boxes variables. The Restore button causes default values to be loaded for geometry and density data in the edit boxes. This can be useful if the user wishes to return to typical values for a projectile. The Cancel button restores the program to the state it was then before the user selected the projectile from the Select Projectile sub menu. In this case, the previous projectile selected before entering the sub menu dialog remains the current projectile.

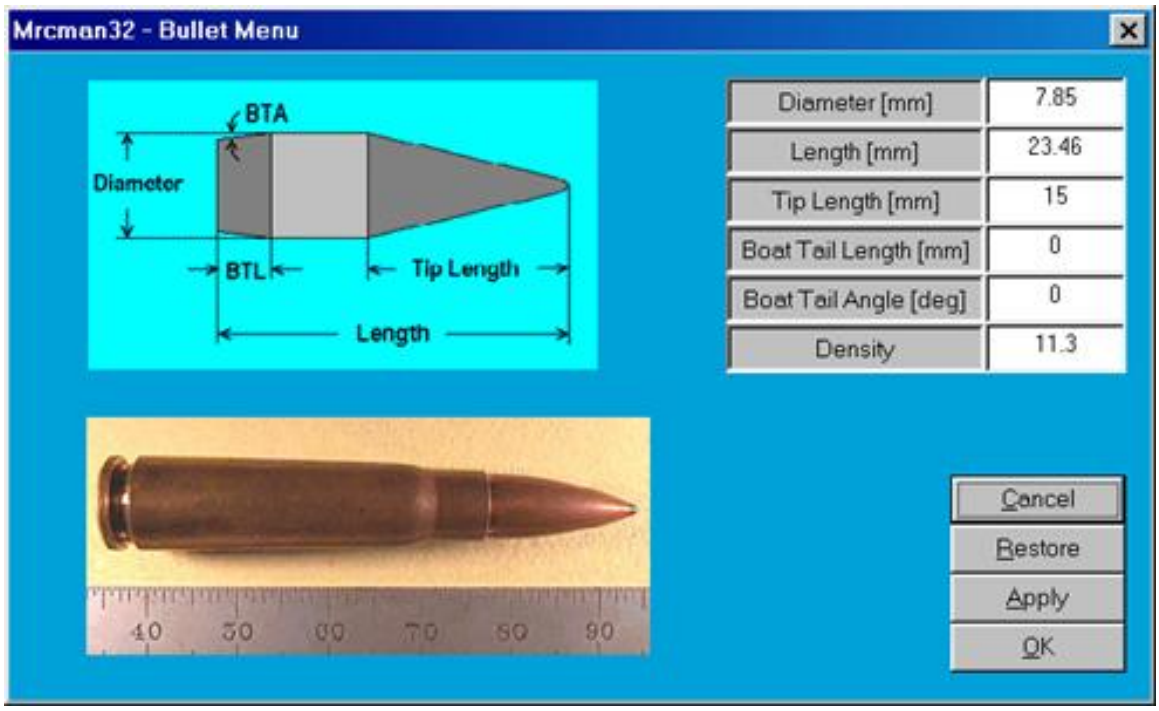

Figure 4-3. Example Projectile Sub-Dialog

4.3 AUTOPSY MODE

Under autopsy mode there is one sub menu item, Shot Line Analysis. Please see Figure 4-5 below. By selecting Shot Line Analysis, Autopsy Mode is entered. As described the previously, autopsy mode is recreation of the original Computer Man developed at Natick. MRCMAN32 began as a complete rewrite of computer man in the early '90s. At the beginning of the rewrite, MRC was delivered a mainframe FORTRAN version of Computer Man as well as a C version created by running the FORTRAN version through a Cobalt Blue FORTRAN to C translator. The reason for the translation was a so that programmers at Natick could add color graphic plotting to the program so that problems could be easily visualized. They accomplished addition of the color graphic routines to view the tissue database, but the program contained unexplained

bugs, its function was not completely understood, and did not have a method for visualization of the shot line. In addition, the original authors of the code were not available. MRC completely re-wrote the computer man in C, using the FORTRAN source listing as a guide. The appearance of the user interface was copied, but original MRC code was for the analysis routines. The only Computer Man remnants existing in MRCMAN32 are the retardation coefficients for the projectiles and the Eyclechymer and Shoemaker tissue database. However, the visible human database and new retardation algorithms were added as recommended user selectable options. The Computer Man retardation coefficients and Eyclechymer and Shoemaker tissue database were kept to compare for legacy applications.

The Eyclechymer and Shoemaker tissue database was also completely reformatted into a whole body model and into a more compact data format and converted to binary to reduce file size and to speed loading from desk. The original computer man Eyclechymer and Shoemaker tissue database was of a monolithic solid man, except one arm and one-leg were missing. Computer man contained algorithms which mirrored the existing arm and leg in order to provide a complete man model. This was probably done to reduce core memory and disk or tape storage requirements for the tissue database, since at the time computer man was developed, storage space was precious and expensive.

Since MRCMan was being developed in the age where memory and disk storage space is inexpensive, it was decided to re-format the Eyclechymer and Shoemaker tissue database into a solid monolithic man with both arms and both legs. This reduced the complexity of ballistic calculations by eliminating the coordinate transformation code required to mirror the arm and leg using the old database format.

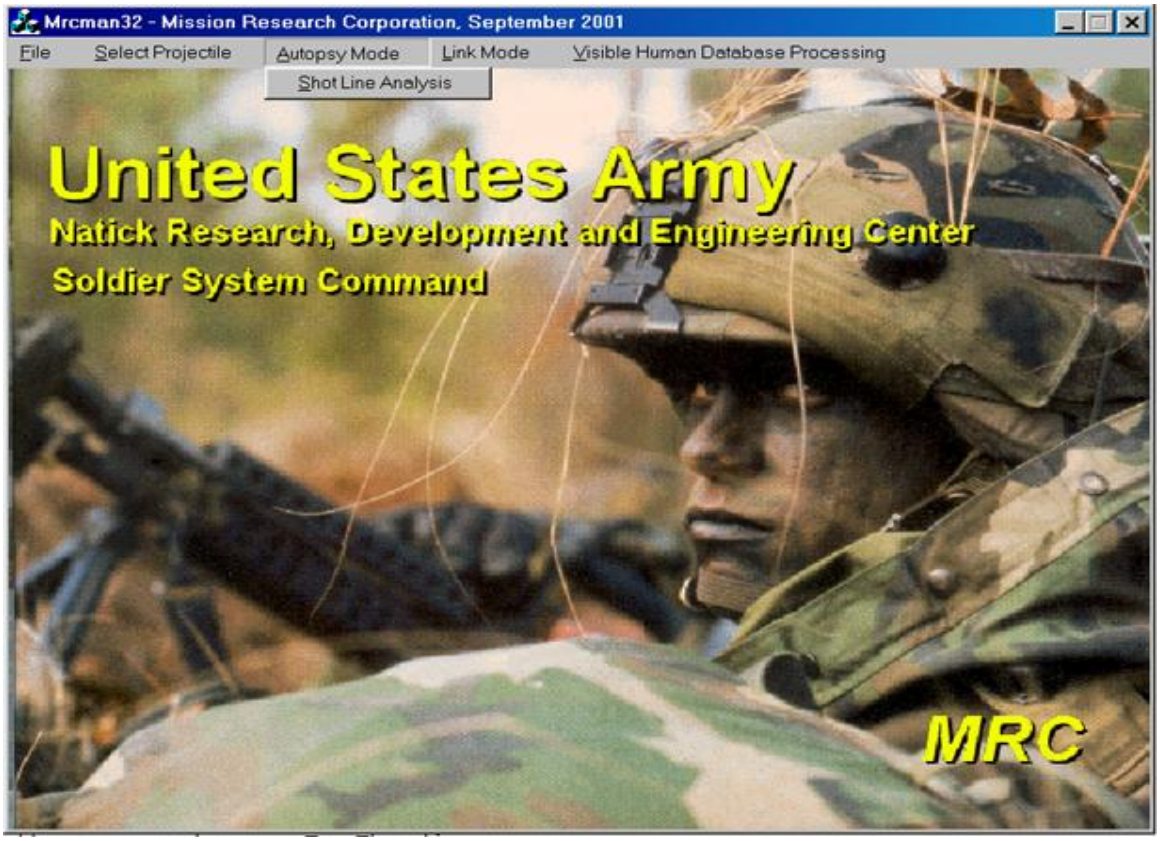

Figure 4-4. Entering Autopsy Mode

Autopsy Mode – Shot Line Analysis Dialog

Autopsy mode determines the speed of a projectile, passing along a straight line through any portion of the human body model from any direction. Autopsy mode shot line analysis generates an XY plot of projectile speed vs. time, in addition to a list of affected tissues and distance traveled through the affected tissue. A casualty criteria code is also assigned to each affected tissue listing in order to aid casualty criteria analysis.

Referring to Figure 4-5, on the left side is the autopsy mode graphic display. There are three views graphically depicting the man model. There is a top view, front view and side view represented by the viewing planes XY, XZ, and YZ. The global coordinate system is oriented such that for the front view, the man faces out of the monitor screen. In this orientation, for the front view, positive X is to the right and negative X to the left. The Y axis runs in the depth direction with positive Y pointing into the screen and negative Y pointing out of the screen. Positive Z is up and negative Z is down.

When Autopsy Mode starts there is a yellow box around the top or XY view of the man. This box represents the current view, which is used to move the shot line in two dimensions using interactive buttons. To cycle the current view through the viewing planes, you may press the window toggle button repeatedly.

The shot line may be moved by two methods. Edit boxes exist for entering the shot line start and end coordinates manually. The start XYZ coordinate defines the location of the start of the projectiles path in the analysis, as the end XYZ coordinate defines the location of the end. The shot line may also be moved by using interactive buttons on the dialog labeled up, down, left and right. The toggle button is used to toggle the moving end of the shot line.

Projectile parameters may be viewed and set at the bottom of the dialog. Remember, projectile type and mass are read-only and are set under the select projectile menu of the MRCMAN32 main screen. Projectile speed may be set by the user in this menu dialog.

The Shoot button is pressed in order to perform Autopsy Mode ballistic analysis. After pressing the shoot button, the plot button may be pressed to display a list of affected tissues along the wound path as well as a two-dimensional plot of projectile speed vs. distance. The Autopsy Mode tissue text file contains a list of affected tissue voxels, tissue numeric type codes, distance traveled through each voxel, as well as casualty criteria analysis codes related to affected tissue type.

In the top left corner of the shot line analysis menu dialog are three edit boxes used to set cutting plane slice locations for the graphic display. By setting these values for autopsy mode, slices of the Eyclechymer and Shoemaker tissue database may be viewed in various orthogonal cutting planes.

The edit box under tissue identification may be used to color the specified tissue white for visual identification in the graphic display. In order to use this the user must have a list of tissue codes. This list may be found in the MRCMAN32 directory, in the tissue sub-directory titled **.\tissue\tisslist.txt**.

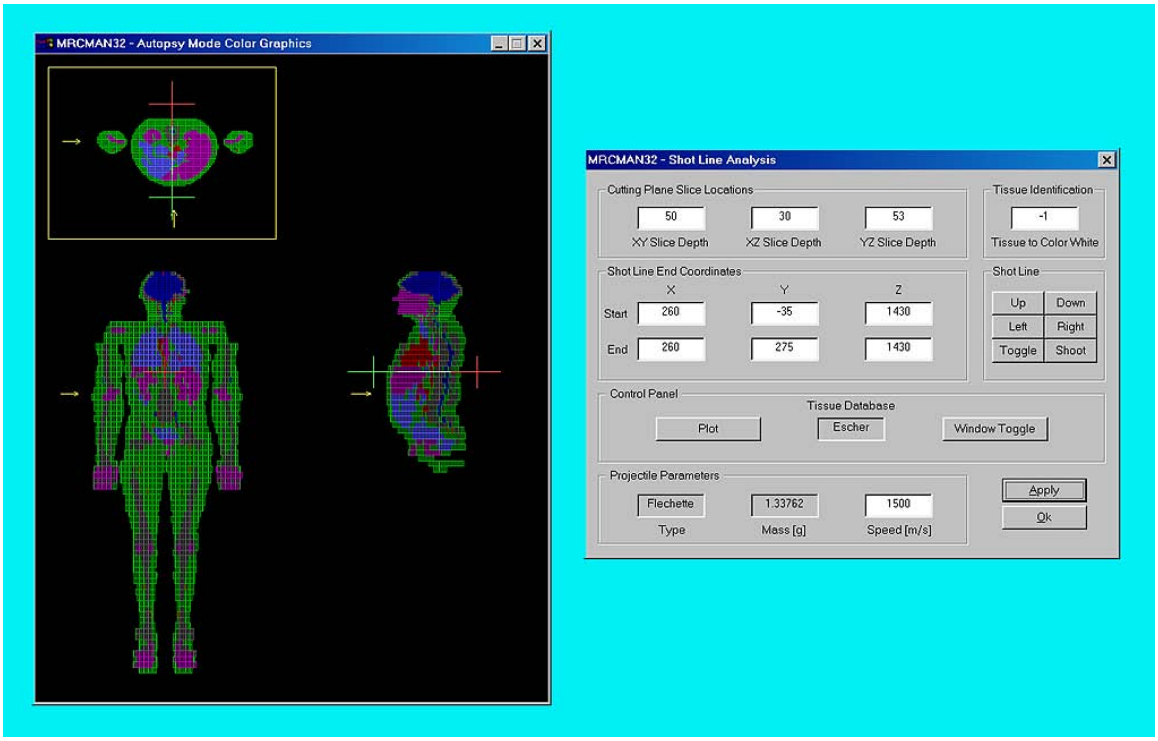

Figure 4-5. Autopsy Mode – Shot Line Analysis Dialog

The buttons at the bottom right corner of the dialog are apply, and okay. The apply button is used to update edit box variables without exiting the dialog. The okay button exits the dialog and returns the user to the MRCMAN32 main menu.

4.4 LINK MODE MENU

Link mode is based on the segmented articulating link model developed by MRC. Under the Link mode menu, sub menu options include Create New Character, Select Tissue Database, Shot Line Analysis, and Vulnerability Analysis.

4.5 CHARACTER SUB-MENU ITEM

When a new Link mode character is created, the tissue database for the appendages must be scaled to the likes of the appendages specified by the user. This is done in the Create New Character dialog, which allows the user to scale the lengths of appendages and the dimensions of voxels for each appendage of the new tissue model in the X,Y and Z directions. Upon exiting the dialog, by clicking the *OK* button, the tissue database is scaled from the "standard man" to the man defined by the link lengths and scale factors specified by the user.

Figure 4-6. Selecting a Link Mode Menu Item

Model link lengths represent a line along the Z axis of the local coordinate systems for each individual appendage. The node at the bottom of each link is used as a reference node and represents the origin of the appendage local coordinate system. The node at the top end of each link represents a point located along the +Z axis and is also used as a reference node. In addition to the top reference node of each link, two other reference nodes are used to define appendage local coordinate systems. One reference node lies at an arbitrary length from the origin node along the +X axis and the other lies at an arbitrary length from the origin node along the +Y axis. These reference notes are used to transform the shot line from the global coordinate system to the appendage local coordinate system, when the shot line is analyzed for each affected appendage.

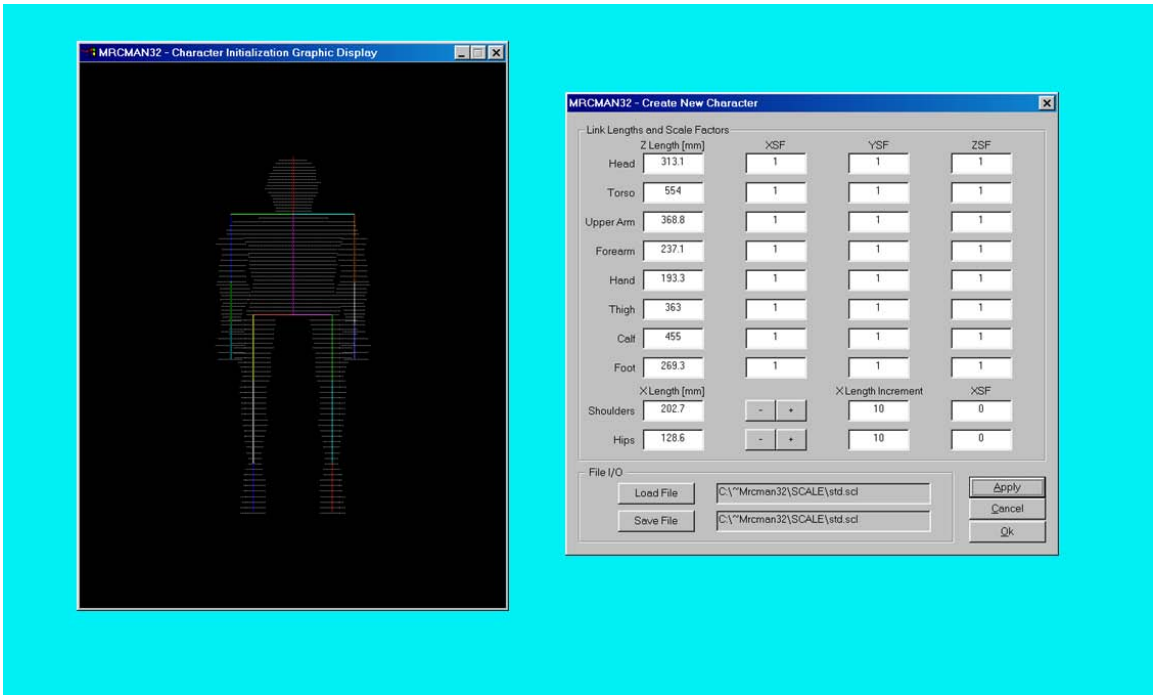

Figure 4-7. Link Mode – New Character Generation Dialog

This dialog is used to set scale factors on the tissue database voxel dimensions for each appendage. Scale factors are used in shot line computations, and determine the boundary planes for each voxel. Voxel X, Y and Z scale factors are set in edit boxes in columns denoted by XSF, YSF and ZSF. Link length edit boxes, while setting the length in millimeters of each link, are ultimately used in the computation of his Z scale factor that is saved in model files along with the X and Y scale factors. The model files are in the MRCMAN32 sub-directory **.\models.**

In the Character Initialization Graphic Display there's a graphic model of the new character and the links representing the articulating node/element model. Changes made in link lengths or to the X or Z scale factors for each appendage are reflected in the graphic model. Changes made in the Y scale factor will not be seen in the graphic model.

Near the bottom of the dialog are edit boxes for the length in millimeters, in the X direction, for the shoulders and hips of the model. Lengths of the shoulders and hips may also be changed by pressing the plus or minus buttons corresponding to either the shoulders or that hips. The increment for changing the length using the buttons may be changed by the edit boxes under X Length Increment. The shoulders and hips may also be scaled in the X direction by using the edit boxes under XSF.

4.6 TISSUE DB (DATABASE)

Link mode operates with two available human body tissue databases, Visible Human and Eyclechymer and Shoemaker. Either of these databases may be selected for use in ballistic analysis by choosing the Select Tissue DB menu item under the Link Mode main menu selection. Selecting the tissue model from the MRCMAN32 main menu configures Link Mode Shot Line Analysis to load the tissue database specified by the user.

In preparation for the development of Link Mode analysis, Beecher Research developed software to segment the Visible Human Database, while MRC developed software to segment the Eyclechymer and Shoemaker tissue database, for use with the articulating link model in MRCMAN32. The result of segmentation are the Visible Human tissue database files, which may be found in the MRCMAN32 directory in .**tissue_Visible_Human\ascii** or .**tissue_Visible_Human\bin_encrypted**. The ASCII tissue database files are encrypted text files containing tissue codes arranged as appendage slices. If an appendage were oriented in the vertical or the local Z direction, the first slice of the file is located at the bottom of the appendage while the last slice is located at the top. The Z axis of the appendage local coordinate system passes through the centroid of the starting and ending appendage slices, the bottom or top voxel surfaces of which, lie in an appendage local coordinate system XY plane.

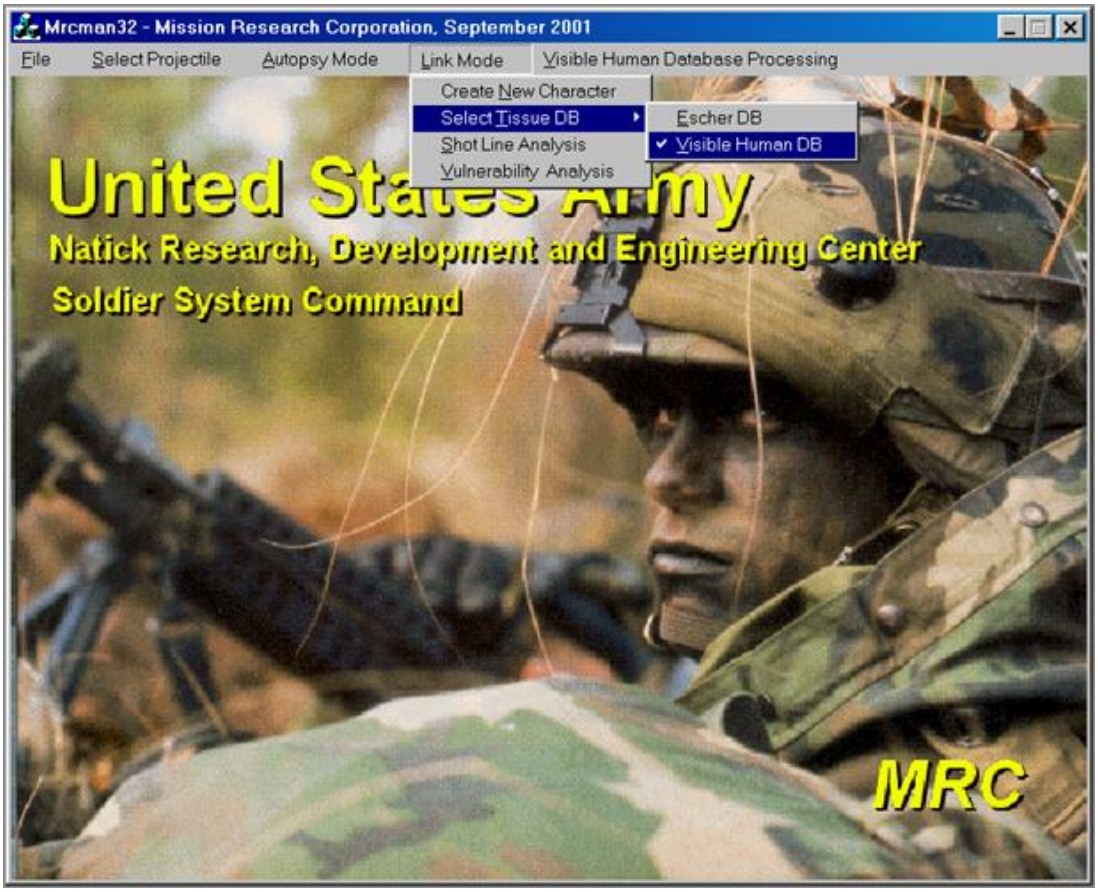

FIGURE 4.8. Link Mode – Selecting the Tissue Database for Use in Ballistic Analysis

The .**tissue_Visible_Human\ascii** directory should be deleted when distributing the code. Binary tissue database files are also encrypted files and should be included for distribution. The corresponding Eyclechymer and Shoemaker tissue database files may be found in the MRCMAN32 directory in **.\tissue**. These files are in ASCII format and not encrypted. Like the Visible Human ASCII files, contain tissue codes arranged as appendage slices. In the .**tissue_Visible_Human or .\tissue** directories, additional sub-directories may be found. The sub-directories contained programs and data used for the development of MRCMAN32 and should also be deleted when distributing the program.

4.7 SHOT LINE ANALYSIS

Link Mode is used to determine the speed of a projectile, passing along a straight line through any portion of the human body model, in an articulated position from any direction. Link mode shot line analysis generates an XY plot of projectile speed vs. time, in addition to a list of affected appendages, tissues and distance traveled through the affected tissue. A casualty criteria code is also assigned to each affected tissue listing in order to aid casualty criteria analysis.

The link man model may be viewed and five different display formats. These are the ellipse point model, the ellipse mesh model, tissue point model, tissue mesh model, and skeletal point model. The ellipse point model is a graphic model created by fitting ellipses, around slices of each appendage. The point model is simply the display of the location of the nodes lying on those ellipses. The ellipse mesh model is the same as the point model except trapezoidal elements are used to connect adjoining nodes.

The tissue point model and tissue mesh model are similar to the ellipse models, except nodes for these models are generated from the tissue database. The skeletal point model is based on the same principle as the tissue models except that only bone and cartilage are displayed.

Referring to Figure 4-9, on the left side is the Link Mode graphic display. There are three views graphically depicting the man model. There is a top view, front view and side view represented by the viewing planes XY, XZ, and YZ. The global coordinate system is oriented matching Autopsy Mode. In this orientation, for the front view, positive X is to the right and negative X to the left. The Y axis runs in the depth direction with positive Y pointing into the screen and negative Y pointing out of the screen. Positive Z is up and negative Z is down.

When Autopsy Mode starts there is a yellow box around the top or XY view of the man. This box represents the current view, which is used to move the shot line in two dimensions using interactive buttons. To cycle the current view through the viewing planes, you may press the window toggle button repeatedly.

The start XYZ coordinate of the shot line defines the location of the start of the projectiles path in the analysis, as the end XYZ coordinate defines the location of the end. The shot line may be moved by using interactive buttons on the dialog labeled up, down, left and right. The toggle button is used to toggle the moving end of the shot line.

Projectile parameters may be viewed and set at the middle of the dialog. Remember, projectile type and mass are read-only and are set under the select projectile menu of the MRCMAN32 main screen. Projectile speed may be set by the user in this menu dialog.

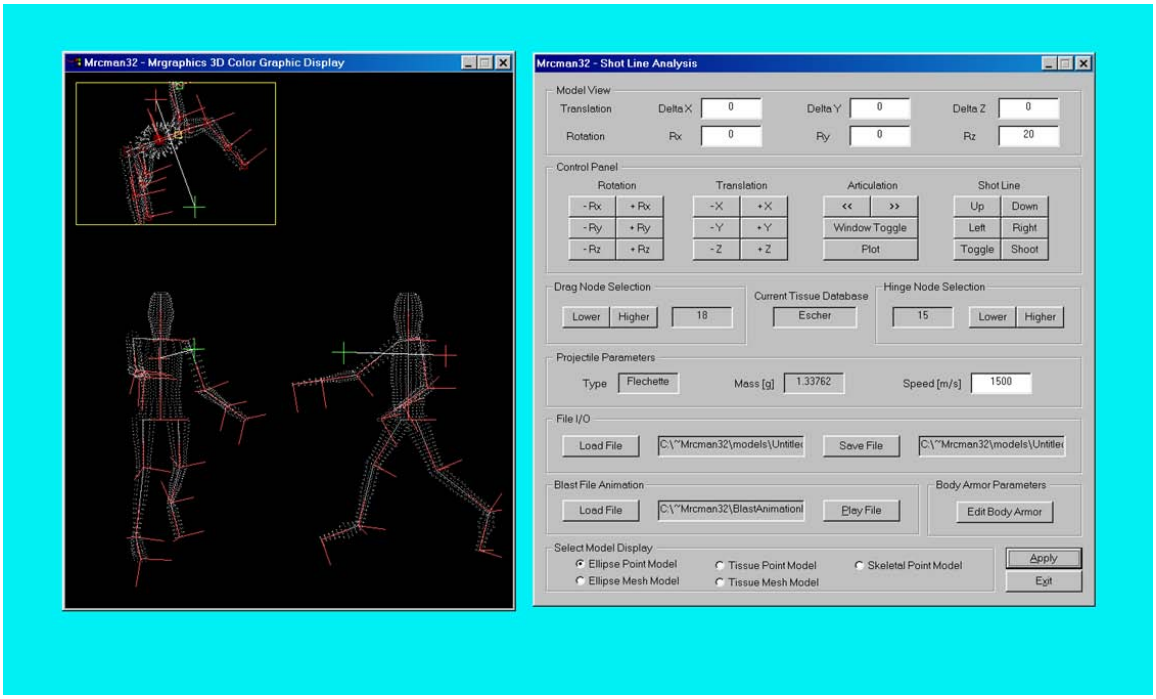

FIGURE 4-9. Link Mode – Shot Line Analysis Dialog

The Shoot button is pressed in order to perform Link Mode ballistic analysis. After pressing the shoot button, the plot button may be pressed to display a list of affected tissues along the wound path as well as a two-dimensional plot of projectile speed vs. distance. The Link Mode tissue text file contains a list of affected appendages and their tissue voxels, tissue numeric type codes, distance traveled through each voxel, as well as casualty criteria analysis codes related to affected tissue type.

At the top of the shot line analysis dialog, are the model view translation and rotation edit boxes. These edit boxes may be used to set the location of the model in three-dimensional space as well as its angular orientation. The model may also be translated an rotated using the buttons under the control panel section of the dialog. These features are used for viewing the link model from different angles. Translations and rotations are of the link model global coordinate system with respect to the world global coordinate system.

The man may also be articulated into various positions and saved to disk file. To articulate an appendage the articulation hinge node first must be selected and then the drag node must be selected. The hinge node, is the node about which the drag node rotates during model articulation. As an example, to raise the Link man's left arm, press the window toggle until the yellow frame is around the side, or YZ view. Next, press the higher button under hinge node

selection until the shoulder node of the left arm is colored yellow. As can be seen, as the higher button is clicked, the yellow hinge node moves sequentially through all of the nodes that make up the Link man model.

Next click the higher button under drag node selection until the green drag node is located at one of the nodes of the left arm. Just as with the hinge node, the green drag node moves sequence away through all of the nodes that make up the link man model. Now under control panel, press the left articulation button (<<) to raise the man's arm. Use this same technique to articulate other link man appendages.

Once the model is in the desired position, orientation and articulation, it may be saved to disk as either in ASCII or binary model file. To save a file press the load file button under File I/O. A dialog will appear, asking the user to specify the file format. The user may also a board to save bite pressing the do not save button. If either ASCII or binary buttons are pressed the Save As dialog will appear, set to the MRCMAN32 models directory. Where it says filename, enter the name of the model. The .mod extension may be added by the user, or if left off, it will be appended by the save routine.

To load a link model file, press the load file button under File I/O. An open dialog will appear, and the model file may be selected by clicking the model name and pressing the open button, or by double clicking the model name.

Blast animation files, generated from the program ATB, may be played in the shot line analysis 3-D color graphic display. To play a file, select load file from the blast file animation section. An open dialog will appear. You may load the blast animation file by clicking the file name and then clicking the open button, or by double clicking the name. Once the file is loaded, it may be played by pressing the play file button. This sequence shall he replay is all of the frames in the blast animation file, pausing on the last frame.

Body armor may be added to the link model ballistic analysis. To edit body armor parameters, click the edit body armor button. The body armor setup dialog will appear. Each appendage has an edit box for setting the retardation factor for the body armor over that appendage. There are also radio buttons which define whether there is no body armor, whether there is body armor as the projectile enters the appendage and whether there is body armor when the projectile exits the appendage. Once body armor is configured click the exit button to return to the shot line analysis dialog.

The buttons at the bottom right corner of the dialog are apply, and okay. The apply button is used to update edit box variables without exiting the dialog. The okay button exits the dialog and returns the user to the MRCMAN32 main menu.

4.8 VULNERABILITY ANALYSIS

In Vulnerability Analysis, or Projection Mode, the articulated link man maybe rotated in three dimensions, and projected on the XZ plane, which corresponds to the screen of the monitor.

Projection Mode was developed to determine the ratio of the exposure area of each appendage projected on the XZ plane to the exposure area of the entire man model.

To enter projection mode, select Vulnerability Analysis from the Link Mode menu, under the MRCMAN32 main menu. This will open the vulnerability analysis dialog as well as the vulnerability analysis graphic display. As stated above the, the graphic display represents the projection of the man model onto the XZ plane. Each appendage is represented by a different color, so that when the window is scanned, the number of pixels represented by each appendage may be counted. The number of pixels in an appendage is divided by the total number of pixels in the man model to arrive at the exposure ratio. Exposure ratios is are depicted in the middle of the vulnerability analysis dialog. To perform the projection analysis, press the scan window button in the lower left hand corner of the vulnerability analysis dialog.

Under model rotation, the model may be rotated to any orientation in three dimensions for exposure analysis. Rotation angles about the various axes may be entered in degrees, in the Enter Rotations edit boxes. Read only edit boxes labeled direction cosines, are the direction cosines of the view plane (the XZ screen plane) vector in the link man global coordinate system. To return the link man to his original position click the reset button at the bottom of the vulnerability analysis dialog.

Pre-articulated link models may be loaded for analysis under File I/O.

In the event the scan of the vulnerability analysis graphic display does not produce exposure ratios, click the color calibration button. This should cause the program to adjust to the computers graphics card.

As with the other dialogs, the apply button forces the update of GUI variables, and the OK button closes the dialog.

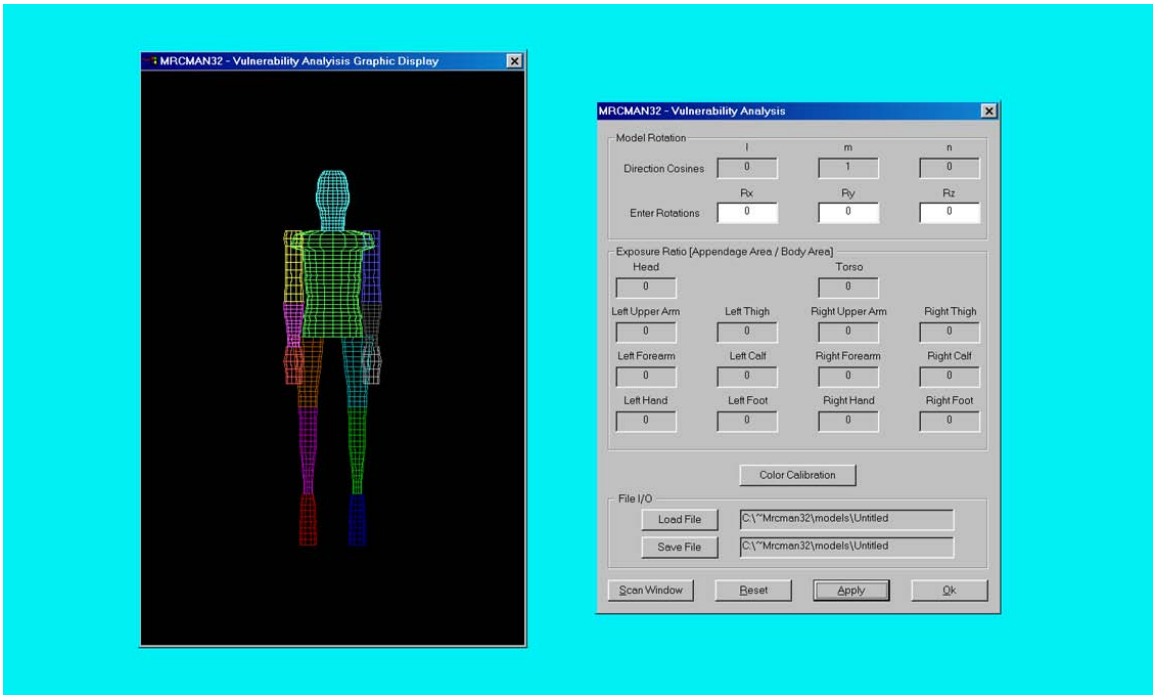

FIGURE 4-10. Link Mode – Vulnerability Analysis Dialog

4.9 VISIBLE HUMAN DATABASE PROCESSING

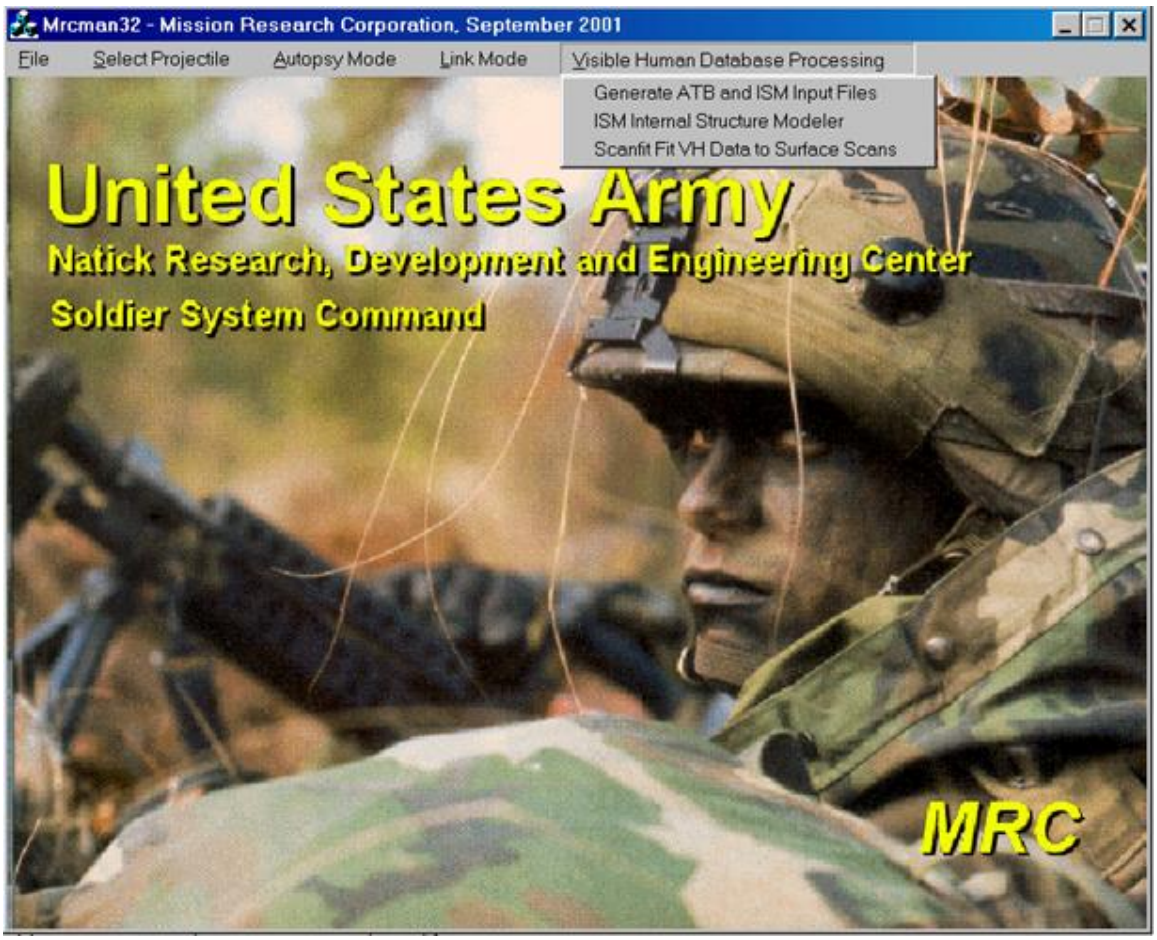

FIGURE 4-11. Shot Line Analysis in Autopsy Mode

Autopsy mode shot line calculations are performed in the function calc_slice_shot_line(), which is in the Autopsy Graphics dialog class (CautopsyGraphicsDlg). A listing of the function is included below. At the top of the function are local variable definitions.

The first operation at the top of function is to compute delta X, Y and Z for the shot line in the link man global corner system. Next, autopsy mode tissue dump the file is opened. This file is located in the database directory under the MRCMAN32 main directory as **.\Database\am_wound.txt**. This file is generated when the shoot button is pressed in autopsy mode along with a plot of projectile speed vs. distance. When the autopsy mode plot button is pressed, the projectile speed plot is displayed along with the **.\Database\am_wound.txt** file.

Next, the t arrays for each component vector of the shot line are zeroed. The tissue database is composed of voxels or volume elements. All voxels are 5 mm by 5 mm on a side by 26 mm in height to, except the head for which voxels are 12 mm in height. Voxels are formed by the intersection of orthogonal cutting planes. Running in the Z direction are XY cutting planes, for

the head, spaced 12 mm apart, and for the rest of the body, 26 mm. Running in the Y direction are XZ cutting planes, spaced 5 mm apart, and similarly in the X direction are YZ cutting planes spaced 5 mm apart. The location of the first and last cutting planes in the global coordinate system axis direction is determined by the location of the link man and the extent of the tissue database. For example, the man is situated facing the XZ plane composing the global coordinate system. To the left of the screen, or to the man's right is the YZ plane. The location along the X axis, of the first YZ cutting plane that defines the tissue database, is the place where if such a plane were slid from the coordinate system origin in the positive direction along X, it would first touch the man. From that point, there is a YZ cutting plane every 5 mm until the last plane touches the man.

The intersection of orthogonal cutting planes, in three dimensions effectively dice is the man up into 5 mm by 5 mm by 26 mm height voxels, except as stated earlier the head which 12 mm high voxels, or XY cutting planes spaced 12 mm apart.

Code Listing A – Shot Line Calculation Top Level Function

```
/*************************************************************************
 Function: Perform Slice Mode (Autopsy Mode) Shot Line Calculations.
 *************************************************************************/

void CAutopsyGraphicsDlg::calc_slice_shot_line()
{
 int num_intersections;
 int tissue_type_string[SHOT_LINE_SIZE];

 int num_t_xy = 0;
 int num_t_xz = 0;
 int num_t_yz = 0;

 double tissue_dist_string[SHOT_LINE_SIZE];

 double t_xy[NUM_SLICES];
 double t_xz[Y_DIM];
 double t_yz[X_DIM];

 double delta_x = (double) (Autopsy_shm.shot_line.end_point[END][X] -
Autopsy_shm.shot_line.end_point[START][X]);
 double delta_y = (double) (Autopsy_shm.shot_line.end_point[END][Y] -
Autopsy_shm.shot_line.end_point[START][Y]);
 double delta_z = (double) (Autopsy_shm.shot_line.end_point[END][Z] -
Autopsy_shm.shot_line.end_point[START][Z]);

 double shot_line[3][SHOT_LINE_SIZE];  /* Shot Line Coordinates in Increasing Order of t */
```

```
 // Initialize the Tissue Text File.

 FILE *fd = (FILE *) fileopen(AUTOPSY_TISSUE_DUMP_FILE,WRITE_ONLY);
 fprintf(fd," MROMEGA - Autopsy Mode Tissue Text File\n\n");
 fclose(fd);

 /* Zero the t Arrays. */

 memset(t_xy,0,NUM_SLICES * sizeof(double));
 memset(t_xz,0,Y_DIM * sizeof(double));
 memset(t_yz,0,X_DIM * sizeof(double));

 if(delta_z != 0.0) num_t_xy = (int) find_t_xy((double *) t_xy,(double) delta_z);
 if(delta_y != 0.0) num_t_xz = (int) find_t_xz(t_xz,delta_y);
 if(delta_x != 0.0) num_t_yz = (int) find_t_yz(t_yz,delta_x);

 num_intersections =
sort_by_t(t_xy,num_t_xy,t_xz,num_t_xz,t_yz,num_t_yz,delta_x,delta_y,delta_z,shot_line);

 // Tissue Text File is Filled in this Function.
 fill_tissue_strings(num_intersections,shot_line,tissue_type_string,tissue_dist_string);

 // Calculate Velocity as a Function of Distance.
 calc_velocity_curve(num_intersections,tissue_type_string,tissue_dist_string);
}
```

The parameter t is a number from 0 to 1 which is equal to the ratio of the distance of a point along the shot line segment measured from the starting point, to the total length of the shot line. In the functions find_t_xy(),find_t_xz(), and find_t_yz(), the t arrays are filled with values of t corresponding to intersections of specified cutting planes with the shot line segment.

The t arrays are then passed into the function sort_by_t(), in which the three arrays are combined and duplicate values of t are eliminated. The XYZ coordinates of the shot line plane intersections are computed and stored in ascending order in an array called shot_line[3][SHOT_LINE_SIZE]. The first dimension of 3 for the shot_line array, allocates room for XY and Z coordinates, while the definition for SHOT_LINE_SIZE is set to value greater than the total number of voxels that can be traversed by the shot line in any shot orientation with the link man. The function then returns to the calling function, the total number of shot line plane intersections.

The coordinates of the shot line plane intersections are then passed to the function fill_tissue_strings() along with the total number of intersections. The fill_tissue_strings() function then computes a point between shot line plane intersections taken in pairs sequentially in ascending order. For example, the first point computed would be the point on the shot line between the first to shot line plane intersections. The second point computed would be a point on the line between the second and third shot line plane intersections and so on. At each of these

computed points, the type of tissue traversed by the projectile is determined and stored in the array tissue_type_string[SHOT_LINE_SIZE], as is the distance between the two points traversed, which is stored in tissue_dist_string[SHOT_LINE_SIZE].

In the final function, calc_velocity_curve(), the speed of the projectile is computed as a function of distance along the shot line. Elements of the tissue_type_string[SHOT_LINE_SIZE] array and the tissue_dist_string[SHOT_LINE_SIZE] array are looped on sequentially. Using retardation coefficients specific to the projectile type, tissue type and the distance through the tissue voxel being traversed, the remaining speed of the projectile as it exits the voxel is computed. A two-dimensional plot of projectile speed vs. distance is generated banking be located in the database directory as **.\Database\am_retard.plt.**

Retardation coefficients for each projectile may be found in the data directory as **.\Data\rcoeff.dat**. On editing the retardation coefficients file, text descriptions of each projectile mark the beginning of the retardation coefficients. Coefficients exist for 10 different general tissue types which are mapped to the many tissue types of either the visible human or Eyclechymer and Shoemaker tissue databases. Below is a table of the tissue type codes and their corresponding meanings. For a complete description, edit the file **.\tissue\TisMapDump_5-26-2000\TissueMappingDump\tismap.dat**.

TABLE 4-1. Retardation Coefficient File Tissue Type Code Descriptions

Tissue Type Code	General Tissue Description
0	Not Used
1	Not Used
2	Vascular Tissue, Internal Organs, Muscle and Nerves
3	Thyroid, Liver, Spleen, Pancreas, Adrenals and Kidney
4	Lung
5	Skeletal Tissue
6	Skeletal Tissue
7	Brain, Eye, Spinal Cord and Vertebral Canal
8	Skeletal Tissue
9	Pharynx, Larynx and Trachea

Shot Line Analysis in Link Mode

Link mode shot line calculations are performed in the function OnButtonShoot(), which is in the Link Mode dialog class (CLinkmodeDlg). A listing of the function is included below. At the top of the function are local variable definitions.

At the beginning of the function, the link mode tissue dump file is opened. This file is located in the database directory under the MRCMAN32 main directory as **.\Database\lm_wound.txt**. This file is generated when the shoot button is pressed in Link Mode along with a plot of projectile

speed vs. distance. When the Link Mode plot button is pressed, the projectile speed plot is displayed along with the **.\Database\lm_wound.txt** file.

Next, the array intersected part[NUM_APPEDAGES] is zeroed, which has the effect of setting all of the elements to false. The intersected part[NUM_APPEDAGES] marks appendages intersected by the shot line and is used further down in the function. The part_order[NUM_APPEDAGES] array is set to on initialized. This array is used to record the order in which appendages are intersected by the shot line.

From Environment.get_environment_ballistics_structure(), the ballistics data structure is filled with projectile parameters such as type, mass, volume, speed and density, as well as shot line parameters. Shot line parameters include start and end XYZ coordinates as well as the shot line direction vector. The function, Character.map_shooting_vars_2_charsim() copies the ballistics data structure to the corresponding character simulator data structure for analysis.

Next, the environment local coordinate systems are copied to the character simulator local award that system variables in the function Environment.update_local_coor_sys_2_charsim(). The final step in setting up the problem for analysis is performed in the function Character.compute_shotline_component_lengths(). In this function delta XY and Z are computed for the shot line segment.

The analysis begins by looping through the appendages to determine shot line/appendage intersections. Whether the shot line intersects an appendage or not, is determined by the function Character_determine_appendage_shot_intersection(). If the shot line is found to intersect an appendage, the appendage number is recorded in the array intersected_part[NUM_APPENDAGES]. The minimum distance from the start of the shot line to the first encountered tissue voxel is also recorded, so that the intersected appendages may be sorted from closest to farthest from the start of the shot line. The appendages are sorted in the function Character.sort_appendages().

Mexico, shot line tissue analysis is performed for each intersected appendage, in the order described above. This is done in the function Character_perform_appendage_shot_line_tissue_analysis(). The file **.\Database\lm_wound.txt**, containing intersected appendages, affected tissue types and casualty criteria analysis codes is generated by this function In the final function, Character.calc_link_velocity_curve(), the speed of the projectile is calculated as it passes through the tissue voxels of the affected appendages. This function also generates a two-dimensional plot of projectile speed vs. distance traversed.

```
////////////////////////////////////////////////////////////////////////////////
// Function: Shoot the MRCMAN.
////////////////////////////////////////////////////////////////////////////////

void CLinkmodeDlg::OnButtonShoot()
{
 unsigned int link;                    // Used to Count Links in for Loop.
```

```
 unsigned int part_count = 0;                  // Used to Count Parts, NOT Links.
 unsigned int intersect_count = 0;             // Used to Count the number of Shotline/Appendage
Intersections.
 unsigned int intersect_flag;                  // Flag to Indicate if the Shotline Intersects the
Appendage.
 unsigned int intersected_part[NUM_APPENDAGES];    // Part Number of the Intersected
Appendage.

 unsigned short int part_cnt;                  // Appendage Count, NOT Link Count.

 int part_order[NUM_APPENDAGES];               // The Parts That Intersect the Shotline
Origin, From Closest to Farthest.

 double temp_min_dist;                         // Temporary Local Variable, Distance from Shotline
Origin to Appendage/Shotline Intersection.
 double min_dist[NUM_APPENDAGES];              // Distance from Shotline Origin to
Appendage/Shotline Intersection.

 POINT_3D local_shot_line_point[2];            // Start and End Coordinates of the Shotline.
 POINT_3D local_shot_line_start[NUM_APPENDAGES];   // Start Coordinates of the Shotline.
Used to Pass Array in Function Call.
 POINT_3D local_shot_line_end[NUM_APPENDAGES];     // End Coordinates of the Shotline.
Used to Pass Array in Function Call.

 DELTA delta;                                  // Length of the Shotline, Projected On to the Global
Coordinate system Axes (x,y,z).

 LINK_SHOT_VARS sub_shot[NUM_APPENDAGES];      // Individual Appendage Shot
Line Info (Num Intersections, Tissue Type and Distance Traveled).

 BALLISTIC_PARAMS ballistics;

 // Initialize the Tissue Text File.
 FILE *fd = (FILE *) fileopen(LINK_TISSUE_DUMP_FILE,WRITE_ONLY);
  fprintf(fd," MROMEGA - Link Mode Tissue Text File\n\n");
 fclose(fd);

 // Zero Arrays Used for Sorting Appendages.
 memset(intersected_part,0,NUM_APPENDAGES * sizeof(unsigned int));

 // Initialize Part Order Array to UNINITIALIZED.
 for(part_cnt=0; part_cnt<NUM_APPENDAGES; ++part_cnt)
 {
  part_order[part_cnt] = (int) UNINITIALIZED;      // UNINITIALIZED = -1
 }
```

```
 // Get a Local Copy of the Environment Ballistics Data Structure.
 Environment.get_environment_ballistics_structure((BALLISTIC_PARAMS *) &ballistics);

 // Map (Copy) Linkmode Variables to Character Simulator Variables.
 Character.map_shooting_vars_2_charsim((BALLISTIC_PARAMS *) &ballistics);

 // Copy the Environment Local Coordinate Systems to Character Simulator Variables.
 Environment.update_local_coor_sys_2_charsim();

 // Compute Delta X,Y and Z for Shot Line.
 Character.compute_shotline_component_lengths((double *) &delta.x,(double *)
&delta.y,(double *) &delta.z);

 // Loop Through Appendages Determining Shot Line Intersections.
 for(link=0; link<NUM_LINKS; ++link)
 {
  // Skip Dummy Parts 8 Through 11.
  if((link < _RIGHT_SHOULDER) || (link > _LEFT_HIP))
  {
   // Check for Proper part_count.

   if(part_count > NUM_APPENDAGES)
   {
    fatal_error("Character_class::OnButtonShoot()","part_count Greater Than NUM_PARTS !");
   }

   intersect_flag = (unsigned int)
Character.determine_appendage_shot_intersection(link,part_count,
                                    (DELTA *) &delta,&temp_min_dist,(POINT_3D *)
local_shot_line_point);
   if(intersect_flag == TRUE)
   {
    // Set intersected_part Equal to the Appendage Number for Sorting.
    intersected_part[intersect_count] = (unsigned int) part_count;
    min_dist[intersect_count] = (double) temp_min_dist;

    // Format Shot Line Data for Function Call Compatibility.
    memcpy((void *) &local_shot_line_start[intersect_count],(void *)
&local_shot_line_point[START],(size_t) sizeof(POINT_3D));
    memcpy((void *) &local_shot_line_end[intersect_count],(void *)
&local_shot_line_point[END],(size_t) sizeof(POINT_3D));

    ++intersect_count;
   }
```

```
   ++part_count;
  }
 }

 // Sort Appendages Along the Shot Line, from Close to Far from Shot Line Starting Point.
 Character.sort_appendages((unsigned int) intersect_count,(unsigned int *)
intersected_part,(double *) min_dist,(int *) part_order);

 // Perform Shot Line Tissue Analysis for Each Intersected Appendage.
 Character.perform_appendage_shot_line_tissue_analysis((unsigned int) intersect_count,(int *)
part_order,(DELTA *) &delta,
                    (POINT_3D *) local_shot_line_start,(POINT_3D *)
local_shot_line_end,(LINK_SHOT_VARS *) sub_shot);

 // Calculate the Projectiles Velocity Profile.
 Character.calc_link_velocity_curve((unsigned int) intersect_count,(LINK_SHOT_VARS *)
sub_shot,(int *) part_order);
}
```

5. NETWORKING AND CHARACTER ANIMATION

From a "black box" point of view, a combat simulator has a means of user input, and a means of outputting data to the user. If the simulator is intended for simultaneous by multiple users, a means must be provided to maintain a consistent representation of the state of the simulation to all of the users. How this is accomplished within the "black box" is subject to constant reevaluation as the available technology changes. However, if the base technology selection is sound and commercially accepted, there is good reason to believe that later changes could be handled as updates rather than major redesign efforts.

A simple example of this is the evolution of Ethernet from 10 million bits per second to 1000 million bits per second. While this evolution requires a hardware update on the individual computers on a network, there has been no fundamental change in the way that software applications "view" the network. The network just transfers data faster.

With an acceptance that technology/design evolution is going to happen, the choices for an appropriate approach are simplified.

5.1 DATA INTEGRITY AND SYSTEM RESOURCES

Any simulation is run for a purpose. Operational requirements are defined by that purpose. In the context of this project there are two classes of applications that are of interest, man-in-the-loop simulations, and deterministic simulations.

Whether the man-in-the-loop simulation is directed towards training or the evaluation of tactics and/or equipment, the key point is that real-time response is critical and absolute, verifiable, correctness is rarely an issue (i.e., if the determination of a shot placement is off by 3cm, it probably doesn't invalidate the value of the overall simulation). If the rotation angle of the right elbow of one participant is not updated for one frame, it is usually acceptable, so long as it is updated in the next frame. However, even in the most chaotic real-time simulation, some discrete events must be handled reliably. The extreme example is the triggering of a nuclear device in the simulation. When it happens, it changes the context of the simulation, it can't be ignored, and it probably won't happen again in the given simulation. Therefore, the entire simulation environment must be apprised to this event, whatever it is, so that all parts of the simulation can respond accordingly.

However, if the simulation is intended for a systematic evaluation of some simulated event, validated and verified results are critical and fast results are important, but real-time operation is of little significance. For example, the faster that a "Monte Carlo" simulation runs, the better.

5.1.1 Objective of Simulation

If the simulation non-deterministic, a certain amount of data loss is usually considered a reasonable trade-off for higher system performance. A deterministic simulation demands that no data be lost. If the nature of the simulation can be performed on a single computer, this is not an issue. However, if the complexity or the scope of the simulation is great enough, it is frequently desirable to distribute the simulation over several computers.

It is the manner in which the various computers are networked, and how this network is used, that is the principal design compromise.

5.1.2 Reliability of Data Transfer

Reliability can be measured in two different dimensions, the delivery of correct information, and the delivery of timely information.

Any form of communications is subject to some level of loss, the issue becomes at what point does that loss become critical and what is an appropriate means of handling data loss. If system "A" sends a message to system "B" that "X" has happened, the message may not arrive at all, or be corrupted beyond B's ability to understand it. This makes two strategies possible, either A just keeps shipping messages, or each message is explicitly acknowledged. This is the basic difference between UDP and TCP packet transfer.

We contend that there are only two general classes of data, 1) continuous state data, and 2) time critical single events.

An example of continuous state data would be the position and joint rotations of a soldier in the simulation, either the avatar of a human participant, or a synthetic soldier. If the driving software (input device or the character simulator), misses updating a frame, the visualization software will render the solider in the last known position/posture. When the new data becomes available, the "soldier" will continue to move normally. The resulting "jerkiness," while not desirable, is acceptable at low enough rates of occurrence. While a secure data transfer could be used for this class of application, the increased reliability would have to be traded off against the increased network bandwidth and latency. For a training simulator, the use of TCP is probably not justified, but for a deterministic simulator it is.

An example of a simple time critical single event is firing a weapon. An M-16, firing 600 rounds per minute, fires 10 rounds per second, or one round every 3 to 15 frames (depending on the visualization system). Hence, it doesn't make sense to ship a recurring message every frame with a flag set that the weapon had been fired, just to cover the possibility that one message had been dropped. Rather, in this case, it is reasonable to send one message and have it explicitly acknowledged that the weapon had been fired.

To provide the flexibility for a general-purpose simulator, every data structure that will be used to transfer data between two systems is assigned the appropriate type of transfer mechanism.

This is set at compile time and is referenced out of the header file for the class of object that will be doing the data exchange. This allows for tuning the application to meet the immediate requirements of future simulation environments without extensive modification.

5.2 NETWORK TOPOLOGY

A line, a star or a ring, how much difference does it make? A few years ago, it could make a lot of difference. Depending on the application, it may still make a significant difference, especially if the system needs to be fault tolerant. However, for the discussion at hand, while a network failure would be annoying, and potentially expensive in terms of lost time, we are not looking at a life critical or a mission critical applications. Rather, we are interested in a cost-effective solution for simulation.

Given the current state of technology that would limit the discussion to 100 Base-T and soon 1000 Base-T Ethernet. The combination of bandwidth and readily available hardware and software support, make it a reasonable first choice to compare any alternatives to.

5.2.1 TCP/IP Based Communications

Irrespective of the physical layer of the communications system, an IP protocol is the most widely supported. A working solution can be had for almost any system, from the most powerful system down to a palm computer. While other protocols could be used or developed, the cost/benefit ratio of such an effort would have to be carefully analyzed.

The IP protocol provides secure transfers, insecure transfers, and multi-cast options.

5.2.2 Alternative Mechanisms

There are several other communications protocols currently available for different target applications, such as network storage and media streaming. But for the raw transfer of data, the only alternative to an IP based system is a form of reflective memory. In a reflective memory system, a dedicated block of system memory (usually dual ported) is installed in each system. These blocks of memory communicate via a dedicated network. When any system updates a location on its local copy, the data is transferred to all of the other systems on the dedicated network and the data is then available to every system at the corresponding location in the local memory space.

While reflective memory was an “up and coming” technology for a while, it has only received limited acceptance. As a result, it is relatively expensive, both in terms of hardware and software. The hardware cost is mainly a question of the economies of scale. Almost every computer has a 100 Base-T interface now, but very few systems have installed reflective memory boards. The other issue with reflective memory hardware is how much memory is enough? If

the simulation requirement grows, to exceed the memory size, even if the bandwidth limit hasn't been reached, the systems will require upgrading.

The software issue is a little more complicated. In an IP environment, the application makes a call and receives a buffer. In a reflective memory environment, the application receives an interrupt and must determine the appropriate buffer to read from. Potentially, the handshaking that can be required is quite complex and memory intensive (up to quad buffering).

So, while reflective memory might be called for in certain cases, it is not considered an immediate design alternative. However, in the event that an application arises that would make effective use of reflective memory, it would be feasible to provide an interface that is compatible with the existing IP calls. This would limit the impact of the modification to the communications library.

5.3 LOAD DISTRIBUTION

A simulation can only run as fast as its slowest component. The main issues are CPU utilization, network bandwidth and memory access. Memory access should be addressed first, since it should never be an issue. A system used for simulations should be provided with enough RAM to avoid using virtual memory. If disk access is needed during run time, background processes should be used to perform them.

5.3.1 Tasks

For the current simulation system, there are XX tasks or processes.

The first process is the server. For reasons of the event process (discussed below) and HLA compatibility, all state data is passed through a single server. It should be noted that the server is subordinate to the event process. Any other process that is updating an object in the simulated environment sends a message to the server with the updated information. The server then passes it on as appropriate. The key phrase here is "as appropriate."

For example, if an event has occurred, such as an explosion, the frame updates from a synthetic soldier animation process would be blocked. The blast effect simulation package in the event processor would then provide all subsequent data until such time as the event process can once more hand off control to the animation process.

The server also serves as the interface to the HLA interface. In HLA terms, the event process "owns" every object that it can modify. That is to say, if an object can be damaged, and the event process is used to determine that damage, then the event process ultimately owns that object. However, temporary ownership can be passed to another process, either within the current simulation, or outside the current HLA Federate, to control rigid body motion of the object.

The second process is the event process. An event is defined as anything in the simulation that can cause non-rigid body behavior of an object in the simulation. In the context of the current discussion, this is nominally thought of as processing for any ordnance which has been set off in the simulation. The event process is sent a message by the process that has triggered the ordnance defining what has been triggered and where it is. The event process is then responsible for all changes in the simulated environment that result from that event.

As a simple example, a rifle is fired. The event process would determine the trajectory and point of initial impact. It then determines if the round was stopped or its new trajectory, as well as the effect on the impacted object. This continues until the round is expended or leaves the simulation. In the event that the round hits a "soldier," a message containing the position, attitude and round trajectory are passed to MRCMan for casualty analysis.

Since the event process determines the new state of any object effected by ordnance, the process can be required to create a significant amount new geometry, with new texture images, as the result of a single event. Transmitting this information to the various user stations is discussed below under the topic of processing requirements versus network bandwidth.

The third is the user interface process. At this time the user process provides simple keyboard and mouse inputs and a visual display. This process is responsible for interpreting using inputs in the context of the objects that the user has control over, formatting messages to the server, character animator, and event process. The data received from the server and event process are then used to render the next visual frame.

The MRCMan process is held on standby until such as the event process determines that a "man" has been hit with some type of ordnance. At this point the joint rotations of MRCMan are set to match the position of the "man" in the simulation. The shot trajectory is set to match the trajectory of the ordnance that caused the event.

The animation process is currently a "Motivate" application. Motivate was developed by The Motion Factory as a game and animation package. It provided a sophisticated set of functions for controlling the mechanical behavior of characters. It should be pointed out that while Motivate simplifies mechanical behavior (e.g., have character #3 pick up box #5) it doesn't provided any AI functions. Hence it may be able to animate a character picking up an object, but it doesn't have any means to determine that the object should be picked up in the first place. At this point, that level of decision making is left to the user.

Since the decision to incorporate Motivate into this project, The Motion Factory has been purchased by SoftImage and the Motivate product is now unsupported.

5.3.2 Computational Work Load vs. Bandwidth

Both CPU cycles and network bandwidth are limited resources. They may both be increasing exponentially, but for any given facility, at any given time, they remain limited. And while the value of graphics cards improves at a phenomenal rate, the demand for increased visual quality

stresses the rest of the system. For example, an AGP interface allows for a greater number of textures to be used in a scene. However, accessing those textures takes up memory and memory bandwidth.

When an event takes place, the event process computes new geometry and textures. But this can be done two ways. Either, the event process can actually determine the final geometry and transmit it to all users processes, or it can determine the critical data and transfer only this data, along with a "seed" for the random number generator, to the user processes. In the first case, the processor load is lower, but the bandwidth requirement is higher. In the second case the processor load goes up, but the network bandwidth is significantly reduced. At this time, the event process is responsible for are model changes and the updated geometry is sent to the user processes. However, future analysis may call for converting to a "local" geometry generation process.

5.4 LATENCY

5.4.1 Application Based Requirement

The time scale of close urban combat is arguably smaller than any other form of combat. The period of time from when something changes (e.g., stepping into a room) through the evaluation of the situation, to taking action, can be a fraction of a second. Therefore, latency is one of the most significant issues in this environment.

For a training system, excessive latency can result in negative training. For a purely analytical application, it doesn't have any significant impact.

5.4.2 Process and Physical Limits

In the current configuration, a user presses a key. As a result, a message is generated. The message is directed at either the animation process (Motivate), or it directed at the event process (a round has been fired).

The animation process takes all available data, updates the current state of each character (the character's mechanical behavior is controlled by a state machine) and checks for interactions between the individual characters and the with the environment. When the revised state is determined, the position and joint rotations are passed to the server, which passes the data to the users. The users then use the current data to render the next frame to be viewed by the user.

Because of the number of interchanges required, and the number of events that could be created per second during a firefight, system performance is currently impossible to project for an arbitrary situation.

5.5 CHARACTER ANIMATION PROCESS

5.5.1 Issues

As mentioned above, the most significant issue at this time is the fact that Motivate is no longer a supported product.

From a behavioral perspective the most significant issue is the lack of dynamics in the motion models. All "moves" are scripted in advance, then inverse kinematics is used at run time to modify the motion to conform to the immediate environment. This results in patterns of motion that are not as realistic as would be desirable.

5.5.2 Process Distribution

Motivate, in its current form, could be run on several machines. Aside from certain operating system issues, there is nothing that would preclude running one copy of Motivate for each character in the simulation.

However, given that the visualization is being run on a Silicon Graphics machine (portable to a Linux platform) and Motivate requires a Windows based system, only a single Motivate process is being used at this time.

6. HIGH LEVEL ARCHITECTURE (HLA) COMPLIANCE

6.1 DATA HIERARCHY

Each object in the simulation is defined by its state. This state is made up of data, the definition of which is dependent on the class of object and fixed, and the geometric data that defines its visual representation, which can change. Via the server, the Federation can obtain any or all of this information.

6.2 COMMUNICATIONS HIERARCHY

As discussed above, during an event, the position, attitude, and geometric description of an object can be changed. Therefore, the head of the logical and communications hierarchy is the event process. The event process controls "ownership" of objects and when, during an event, the process forces the return of ownership, it can radically alter or eliminate an object.

The next element in the communications hierarchy is the server. It sole task is to accumulate and distribute information. The sources of information are the event process, Motivate, the Federate/Federation interface, and the user process (for a more sophisticated man in the loop application).

The Federate/Federation interface serves as a translator and gateway to the Federation. It maintains the data appropriate to determine what, if any, data available on the server should be sent out to the Federation, and what data available in the federation is used to update the server.

6.3 INTEROPERABILITY ISSUES

The single most significant issue is the geometric data. The update of the geometry is the point project, but the variation in models from one simulation environment to another makes it pointless to pass the data outside the immediate simulation. As an example, consider an urban combat training simulation with helicopter support. First, it is extremely unlikely that the flight simulator is using model database that has a level of detail suitable for in infantry simulation. So if something is damaged, there is no one to one correspondence between the geometry of the two simulations.

This document reports research undertaken at the U.S. Army Natick Soldier Research, Development and Engineering Center, Natick, MA, and has been assigned No. NATICK/TR-10/007 in a series of reports approved for publication.